UCRL-MA-151338

HADES User's Manual

Maurice B. Aufderheide, Dale M. Slone, Alexis E. Schach von Wittenau

December 26, 2002

U.S. Department of Energy

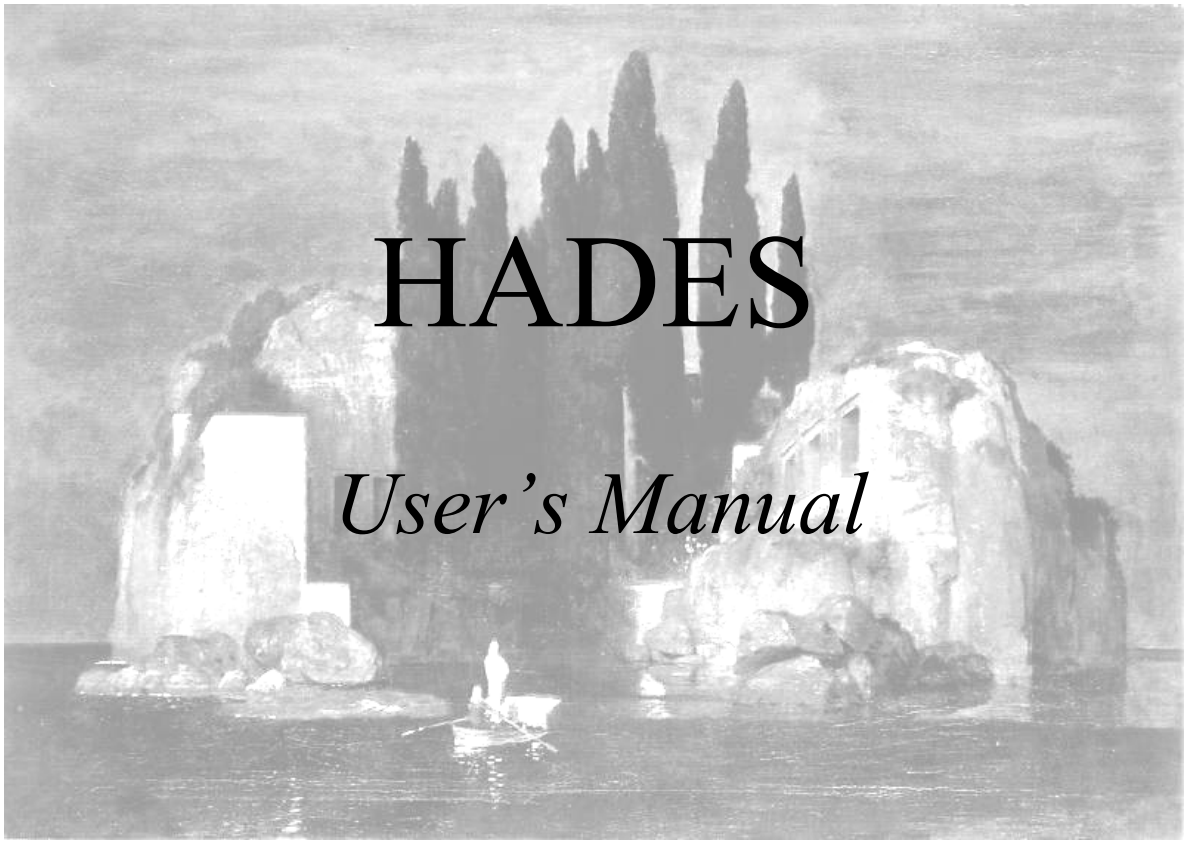
Lawrence
Livermore
National
Laboratory

DISCLAIMER

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

UCRL-MA-151338

‘Αιδης



HADES

User’s Manual

Maurice B. Aufderheide III
Dale Slone
Alexis Schach von Wittenau

"This is the last word that Ajax speaks to you. The rest he will tell to the shades in Hades."

Sophocles Ajax 864-865

"*Sheol beneath is stirred up to meet you when you come, it rouses the shades to greet you, all who were leaders of the earth; it raises from their thrones all who were kings of the nations."*

Isaiah, 14:9, RSV

Background picture on the title page:

The Isle of the Dead

Arnold Böcklin

1880

Oil on Canvas

Kunstmuseum Basel, Basel

Table of Contents

Table of Contents **4**
Purpose **7**
Running the Code **7**
Why the name? **7**
Acknowledgements **8**
Image Formats **8**
A Note on Terminology **8**
A Brief Primer on Radiography **9**
Magnification 9
Blurring 9
Units: What is returned in a HADES image? 10
Options For Specifying Radiographic Probes **11**
Overview 11
X-Ray and Neutron Options 11
1. Specifying Mass Attenuation Coefficients *11*
2. Specifying a Monochromatic Source *11*
3. Specifying a Spectrum *12*
4. Spectrum and Full Detector Response *12*
The Proton Option 12
1. Perfect Lensing Approximation *12*
2. Inclusion of Proton Beam Line *12*
HADES Coordinate System **13**
HADES Keywords **14**
Format of the input deck *14*
Syntactical Conventions 16
General Body Commands 16
Positioning: *16*
File Commands *17*
Control Commands *19*
Mesh Manipulation Commands *19*
Source and Detector Commands *21*
Background/Scattering Commands *23*
Block Commands 24
Probe Blocks *25*
Proton Block Commands 25

Spot Block Commands ... 26
Solid Body Blocks ... *27*
Commands Used in All Solid Body Blocks ... 27
Plate Block Commands ... 29
Sphere Block Commands ... 29
Ellipsoid Block Commands ... 30
Cone Block Commands ... 30
Collimator Block Commands ... 30
General Collimator Block Commands ... 31
ZRProfile Block Commands ... 32
Torus Block Commands ... 33
Excluded Object Block Commands ... 33
Mesh Rotation Blocks ... *35*
Multiangle Commands ... 35
Multiphi Commands ... 35
Multitheta Commands ... 36
Multi_thetaphi Commands ... 36
Sinogram commands ... 37
Miscellaneous Blocks ... *37*
Camera Block Commands ... 37
Specification of Materials in a Problem ... 38
Old-style Collimator Commands ... 39
Appendix A: Example Input Decks ... 40
1. Nested spherical shells using solid body objects and no mesh ... 40
2. Nested spheres using excluded objects and detector parameters ... 41
Input Deck and Radiograph ... *41*
Spectrum File Used ... *43*
Detector Files ... *44*
DQE File: bgo_2cm_dqe_example2.dat ... 44
Blur File ... 44
3. Neutron radiograph of a meshed object ... 45
Input Deck and Radiograph ... *46*
1DrASCII Mesh File example3.mesh ... *46*
4. Perfect Lensing Proton Radiography for a 1DrASCII mesh ... 46
5. Proton Radiography Using the Brookaven National Laboratory U Line ... 47
Input Deck and Radiograph ... *47*
Beamline Specification Deck ... *48*
6. Example of Generalized Collimators and a Mixture by Weight % ... 49
Appendix B: Reading .sdt Files with IDL ... 51
Appendix C: Detector Definition ... 53
Appendix D: Post Processing ... 56
Appendix E: Other HADES Invocations ... 57

Restarts 57
Parallel HADES 57
Appendix F: Photon Dose Conversion Curve 59
Appendix G: Data Used and Locations 61
Location of data files 61
Index of HADES Keywords 62

Purpose

HADES is a computer code that simulates transmission radiography through a mesh and/or solid bodies. This code is a successor to an old code called XRAY. HADES is designed to be backward compatible with XRAY (i.e. it will accept and correctly parse XRAY input decks), but HADES has many more features that greatly enhance its versatility and its simulation of experimental effects. HADES can simulate X-Ray radiography, neutron radiography or GeV proton radiography using ray-tracing techniques. A short article has been published which discusses HADES' capabilities[1]. HADES also has some capability for simulating pinhole imaging and backlight imaging. The best way to obtain HADES is to ask one of the authors for a copy. This manual covers all features in version 2.5.0 of HADES.

Running the Code

Currently, HADES requires an input deck. The code runs using the following style of invocation:

hades *deckname* [*meshname*]

deckname is the name of the ASCII input deck HADES is to use for setting up the radiographic parameters of the image. These parameters include the geometry, detector size, source and detector properties, solid bodies to be included, and mesh properties.

meshname is the name of the file which contains the information on the mesh which is to be radiographed. If no mesh is to be radiographed (i.e. only solid bodies are being simulated), the second argument should be blank.

In addition to these files, HADES may require the use of other files, depending on what the user is requesting. These files contain information about the materials in a mesh, detector or source spectrum information, and parameters for detailed modeling of a proton radiography beamline.

Why the name?

In Greek mythology, Hades was the underworld, the abode of the dead. In some English translations of Greek classics and the Bible, the dead are referred to as "shades", that is, shadows. Radiography is the use of shadow images created by attenuation of radiation. Hence the name. Also, it has been a heck of a lot of fun to build this code!

[1] M.B Aufderheide, D. M. Slone and A. E. Schach von Wittenau, "HADES, A Radiographic Simulation Code", in *Review of Progress in Quantitative Nondestructive Evaluation,* **20A**, AIP Conf. Proc. 557, (2001), pp. 507-513.

Acknowledgements

The earliest version of HADES was the result of a summer project (called TEXRAY), executed by James Ayers. Some options within HADES have been motivated by work with Allen Mathews of Los Alamos National Laboratory. Yi-Ming Wang of LLNL developed some of the multi-angle code in HADES. We also thank Harry Martz and Jessie Jackson of the Center for Nondestructive Evaluation at LLNL for suggestions and requests. MBA thanks Joseph Gray of the Center for Nondestructive Evaluation at Iowa State University for fruitful discussions.

Image Formats

HADES can presently produce images in PGM format, TIFF format, Silo format, and View format. The default format support by HADES is the "View" format, which is used by an LLNL image processing code called View. This format is simple enough that programs such as IDL and NIH Image can easily be trained to read it. (See "Appendix B: Reading .sdt Files with IDL" for more discussion of how to read files with IDL.) The image is sent out in a binary piece (with suffix "sdt") which is a floating point raster scan of the image, and an ASCII piece (with suffix "spr") which contains the "header" information about image size and format.

The `preview` option produces the image in raw portable graymap (PGM) format, which can be looked at with *xv* or other image packages. One note of caution: this PGM file has been scaled so that the range of the image is stretched between 0 and 255. The original numbers can be recovered by looking at the comment on the second line of the PGM file, which lists the original min and max of the file. A similar caution applies for the TIFF format: HADES scales the floating-point image so that it fits on the required 8-bit integer TIFF format. Thus PGM and TIFF images will look exactly like the View image, but the exact absolute values within the pixels will have been lost.

A Note on Terminology

A HADES "keyword" is a word or phrase that is used in an input deck to control the operation of the code. It always starts a new command. In this manual, keywords are always set in `Typist font`, so that the user knows they are keywords. An example is the discussion of the `preview` option in the previous paragraph. All HADES keywords are listed alphabetically in the Index on page 62.

HADES simulates radiographs of two general kinds of objects: meshes and solid-body objects. A mesh is an object whose density and composition variation is recorded in a set of 1D, 2D, or 3D zones. A solid-body object is an object of uniform density and composition, whose volume is enclosed by a set of surfaces.

A Brief Primer on Radiography

Magnification

If one sets aside the complications of radiation interaction with detectors/film, and the issue of scattering, radiography is a very simple experimental technique. Typically, the radiation originates from a source that we will treat as a point for now. A cone of radiation passes through the object of interest, suffering attenuation, and is detected on a plane behind the object. Figure 1 shows this geometry.

The object is a distance *l* from the source. The object plane is that plane, perpendicular to the radiographic axis, which bisects the object. The detector plane is that plane, perpendicular to the radiographic axis, which coincides with the detector. The detector plane is a distance *l'* from the source. It can be seen by similar triangles that any feature of height *h* in the object plane will have a larger height *h' = l'/l h* at the detector plane. The ratio *l'/l* is referred to as the magnification *M* of the system. It can also be seen in the above figure that the magnification of all pieces of the object is not exactly *M*, since the object is of finite width. However, typically the width of the object is small compared to *l'*, so that *M* is a good approximate magnification for the whole object. In simulations, *M* as such is not used by the code, but it is a useful quantity for estimates.

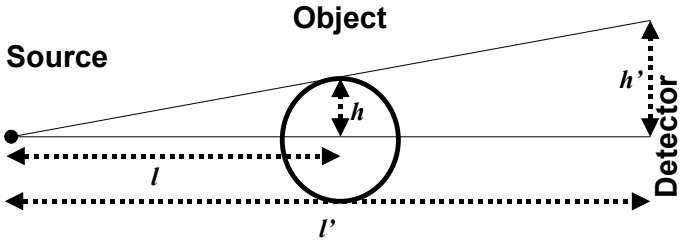


Figure 1: Radiographic Magnification

Blurring

All X-ray and neutron sources have finite extent and this leads to a blur in the final image. Figure 2 illustrates this phenomenon. Let *b* and *b'* be the spot size and its projection through the object. Again using trigonometry, it can be seen that *b' = (M-1)b*. For the case shown here, *M* is greater than 2, so that *b'* is larger than *b*. It can be seen that the radiograph for such system will be a convolution of the spot profile through the object. Often, this blur is treated as a magnified image of the spot convolved with the image at the detector plane. While this is a useful approximation, it is not strictly correct, because the penumbra of each feature in the object has a slightly different magnification.

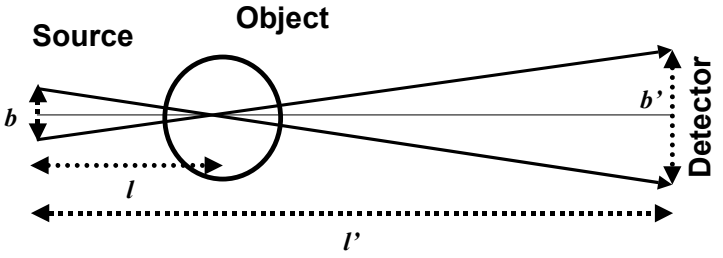


Figure 2: Radiographic Blur from a Spot of Finite Size

Detectors also have blurs. If we assume a point source, this blur is a convolution of the detector blur function with the perfectly resolved radiograph at the detector plane. A radiograph with a finite spot and a true detector will be the result of a convolution of the spot profile through the object convolved with the detector blur at the detector plane. Because of

the differing nature of these two convolutions, these blurring operations strictly do not multiply in Fourier space. However, as a first approximation one can act as if they do. If we further assume that each blurring function is a Gaussian characterized by a full-width at half maximum (FWHM), then the total blur can be approximated by

$$FWHM_{tot} = \sqrt{FWHM_D^2 + (M-1)^2 FWHM_s^2}$$

where the *D* and *s* subscripts stand for detector and spot, respectively. This total blur has been computed in the detector plane. To project back to the object, divide $FWHM_{tot}$ by *M*.

The discussion in this section has been simple in its treatment of radiographic blur. HADES can currently treat the physics in a much more sophisticated, albeit complicated, fashion. For example, HADES can model energy-dependent detector blur functions, which are convolved with the image in an energy dependent fashion. HADES can also model finite spot size by integrating over the spot (requiring parallel operation of the code).

Units: What is returned in a HADES image?

In a radiography experiment, the detector measures whatever energy is deposited in it. This includes both scattered radiation and unscattered, attenuated radiation. The dynamic range of these intensities can be more than a factor of 1000, depending on the field of view. In reality, detectors are challenged when covering such wide ranges. Also, a detector may have responses that are not linear with respect to the deposited energy.

At any rate, this is not what HADES returns in a "radiograph". For each detector pixel, HADES computes λ, the pathlength, from the source to the center of each pixel in the detector. The pathlength is given by

$$\lambda = \int \mu \rho \, dl$$

where μ is the mass absorption coefficient (roughly 0.03 cm^2/g for 4 MeV photons in iron), ρ is the density at each point along the ray, and the integral is along the path of the ray. Note that λ is unitless. If I is the transmitted intensity and I_o is the incident intensity, the pathlength through the object and is equal to $-ln(I/I_o)$ for this pixel, for monochromatic photons, if there were no scattering. Note that I and I_o include only unscattered radiation. A HADES "radiograph" is then an image of the pathlength $\mathcal{R}$ through the object. If one needs the scatter profile included in the radiograph, a Monte-Carlo code will have to be used. Luckily, for many applications, the scattering profile is not a very large perturbation to this image, so that HADES is adequate. Also, if a scattering profile is needed, it can be computed for this system and added to the HADES image. HADES always returns images whose units are pathlength. This is done because most tomography codes want radiographs in these units. For linear detectors, the HADES simulation $\mathcal{R}$ needs to be converted to $e^{-\mathcal{R}}$ for direct comparison with experiment.

If statistics are included in the simulations, the I values for each pixel (in units of numbers of photons/pixel) are attenuated appropriately for this ray and then perturbed using Poisson statistics. These perturbed I values are then divided by I_o. If blurring has been selected by the user, the image is blurred at the detector (in terms of I) plane before statistical fluctuations and then converted back into units of $-ln(I/I_0)$. The bottom line is that, whatever choices are made about the radiation source or the detector response, the final radiographs are given in units of $-ln(I/I_0)$.

Options For Specifying Radiographic Probes

Overview

HADES allows the user wide latitude in specifying radiographic probes. Presently, the user can specify X-Rays, neutrons and GeV protons. For X-Rays and neutrons, there are 3 ways of specifying the spectral content of the incident beam, of increasing complexity (and realism). For protons, there is only a monochromatic mode. In this section, we summarize these options and how the user can set them. For the X-Rays and neutrons, we start with the simplest option and proceed to the most complex.

X-Ray and Neutron Options

1. Specifying Mass Attenuation Coefficients

The user uses the `mac` command to specify a mass absorption coefficient (in units of cm^2/g) for each material in the mesh and for each non-mesh object. The user does not have to specify isotopic compositions, but does have to come up with correct coefficients for each material somehow. The user can include statistical fluctuations by using the `XRay intensity` option, which specifies the total number of X-Ray photons uniformly spread over the field of view. This option is rarely used because it requires some expertise about how to compute effective mass absorption coefficients. HADES can compute these using its own data tables as shown in the next three options.

2. Specifying a Monochromatic Source

This option is the best compromise between ease of operation and fidelity to the true Physics of radiography for a casual user of the code. The user specifies a single X-Ray energy using the `gamma energy` or `neutron energy` option. The user also must use the `naminp` option in order to specify a file with the isotopic composition of each material in the mesh. Within this file the `isofrac` command is used to specify material composition. The `isofrac` command must also be used to specify compositions of all solid-body objects. Note that HADES will properly mix all material components, so that the user need not do this. The `Dose` or `XRay intensity` option can be used to include statistical fluctuations on the image. If `Dose` is specified, the code converts this dose into number of photons per pixel for the given photon energy. If `XRay Intensity` is specified, the number of photons per pixel is computed directly from this number and used for fluctuations. See Example 1 in Appendix A: Example Input Decks for an illustration of this option.

3. Specifying a Spectrum

Here `spectrumfile` is used to give the name of a file containing information about the incident photon or neutron spectrum. In this file, the user specifies the endpoint energies of each spectral bin, the dose in that bin, and a relative detector sensitivity. The code uses these energies to make conversions from dose to relative number of photons, including the specified relative sensitivities. It performs averages over each bin for determining cross sections (from the LLNL Evaluated Photon Data Library) for that group. If `Dose` is specified, the code parcels this total dose among the bins, working out statistical fluctuations in each bin after transmission. The goal of this option was to allow the user more flexibility in defining spectra and detector properties. This option has been successfully used for low energy X-Rays and higher energy sources. The user can also supply an input spectrum in units of number per bin rather than dose per bin using the `number spectrum` option. Also, if the user prefers to specify a differential spectrum (i.e. dN/dE or dDose/dE), HADES will accept this if the `per energy` flag is set.

4. Spectrum and Full Detector Response

In this treatment, the full spectral information of the previous approach is used, plus a more accurate depiction of detector response to radiation. Detector response is characterized with a file which tabulates detector blur and another file which tabulates detector quantum efficiency (DQE) as functions of energy. These tabulations are input using the `detector blur file` and `dqe data file` commands, respectively. These files can be used singly, or together. See Example 2 in Appendix A: Example Input Decks for an illustration of this option.

The Proton Option

1. Perfect Lensing Approximation

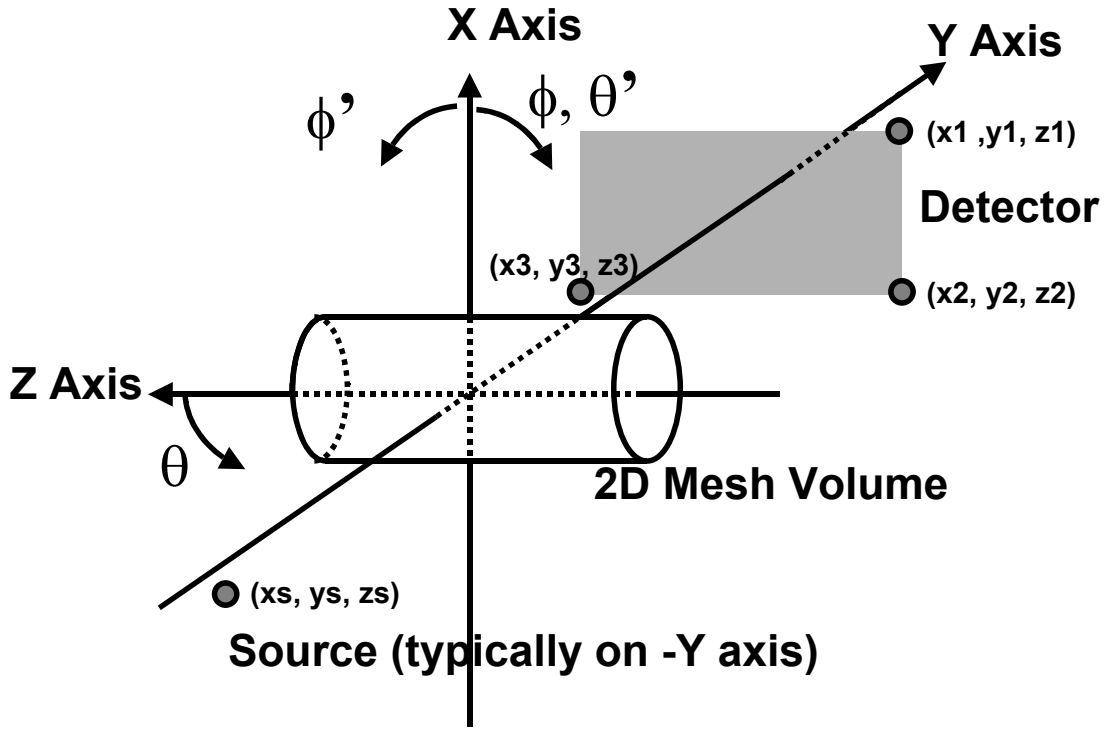
Presently the user specifies one incident proton energy and two angular cuts (that is, two maximum angles for transmission through collimators). Perfect lensing is assumed. When this option is specified, many files are sent out by the code. The default root file name is “pray”. The most important images are the final images from each angular cut which are called praycut1 and praycut2. If the user specifies an intensity, “n” is appended to these names. These images are, as always, $-ln(I/I_0)$ images, with nuclear attenuation and angular attenuation included, plus statistical fluctuations, if an intensity has been specified. Many more images are produced of the intermediate results, but these are mainly of use to the code developer or the advanced student of proton radiography and will not be discussed in this section. See Example 4 in Appendix A: Example Input Decks for an illustration of this option.

2. Inclusion of Proton Beam Line

In this more detailed and realistic treatment of proton radiography, the proton beam is traced through the beamline, including quadrupole magnets and collimators. The angular and spatial distribution of protons exiting the object are propagated from the object center plane. Each ray is broken into beamlets which are numerically integrated through the system. Beamlet pieces which hit lenses or collimators are killed, without the production of secondary particles. All contributions are recombined at the detector planes. This treatment allows the

user to study the effects of object misalignment and magnet tuning on the calculated images. See Example 5 in Appendix A: Example Input Decks for an illustration of this option.

HADES Coordinate System



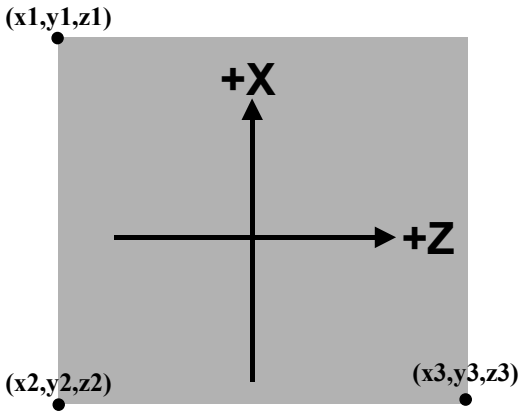
All conventions use a right-handed coordinate system.

All solid body objects are defined relative to the HADES origin. The (0,0,0) point of all mesh volumes is also located at the HADES origin. The HADES detection plane is defined by the points (x_1,y_1,z_1), (x_2,y_2,z_2) and (x_3,y_3,z_3). Although in principle these points can be arbitrarily oriented and need not define a rectangle, it is worth noting that much of the HADES code was written with the assumption of a rectangular (or square) detector, with the detector plane being perpendicular to the y axis. HADES starts at (x_1,y_1,z_1) and rasters to (x_3,y_3,z_3). An easier way to specify these detector points is to use the `camera` command, as discussed on page 37. The camera command specifies problem geometry in a much simpler way.

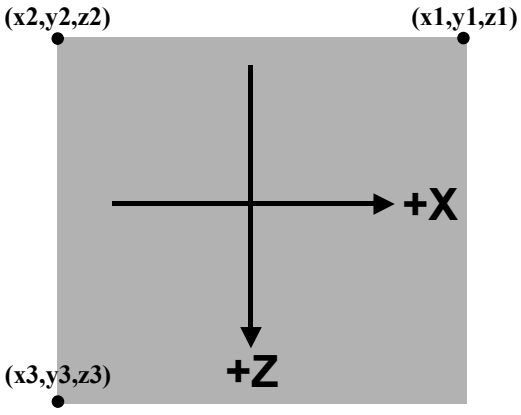
HADES uses the above coordinate system, which is designed so that the default source to detector axis (sometimes called the "radiographic axis") is the Y Axis. In this way, a 2D mesh (assumed to have the Z Axis as the symmetry axis) is orthogonal to the radiographic axis. For rotations, HADES currently has three angle sets. In the default angle set (`ORIGINAL`), the θ rotation is first and it is about the X axis. ϕ is second and it is about the Z axis. In the newer sets of angular orientations (`CTX, CTY`), (denoted by the primed angles on the figure), the θ' rotation is first and it is about the Z axis. ϕ' is second and it is about the (X, -Y) axis.

The newer angle sets are useful for computed tomography simulations, where one typically takes views on the X-Y plane around an object and may rotate off this plane if necessary.

This coordinate system is optimal when considering meshes with 2D symmetries. For 3D meshes, this system may not be optimal, but has been chosen so as to be consistent with the 2D meshes. The user can pick a better orientation of the mesh by using nonzero θ and ϕ values.



Default Orientation



CT Orientation

The final radiograph is presented in an orientation as shown above. The orientation at left is the default orientation and is as if you were viewing the radiograph from behind the detector toward the source.

It is also possible to use the `ORIENTATION` command to set the radiographic orientation to be more appropriate for computed tomography; i.e. with the z-axis vertical. This orientation is depicted at right. For more information, see the `ORIENTATION` command write-up.

HADES Keywords

Format of the input deck

When HADES reads an input deck, it looks for the keywords that specify the problem. This method of parsing the input deck allows any ordering of the commands. Error checking for invalid entries is done after the entire deck is read.

Commenting is allowed in the input deck. The comment format is the standard C++ format, using two slashes (//) followed by the comment. Whenever two slashes together are found on a line, all of the line after the two slashes is ignored by the parser.

Multiple keywords are allowed on one line. The only rules are that the keywords may only accept one parameter and that the line length may not exceed the maximum (currently set at 399 characters). In general, it is not advised to put unrelated keywords on the same line, but the three coordinates of a point may easily be placed on the same line. Spaces are the only

required delimiters. Semicolons are also useful delimiters. In lines with multiple keywords, DO NOT mix delimiters.

Equal signs are allowed in the input deck, but they are not required. A keyword directly followed by data will be parsed correctly; an equal sign between the data and the keyword is ignored and has no effect whatsoever.

In general, random text in the input deck (text which is not between a keyword and its argument) is allowed, but woe to the user! If you do not comment any random text out of the file, the parser might be confused. For example, the line

`xs` 23.1 used on Thursdays only

has TWO keywords: `xs` is assigned to 23.1, and `ys` (at the end of the word Thursdays) is confused. The behavior of the input deck parser is not completely predictable; therefore, it is a good idea to put in comments; the above line should look like this:

`xs` 23.1 //used on Thursdays only

No confusion would result from the line with a comment. This feature of the parser is the main source of untrapped HADES errors. Be particularly careful of commands that read in file names. If the file name has any HADES keywords in it, that keyword will get reset in unpredictable ways. For example, if the user were specifying the name of the output file kernel to be “morryscalc”, the command would be:

`outfile` morryscalc

The output file kernel will be set to “morryscalc”. Unfortunately, `ys` will also be set to a random value, probably 0.0. If this command occurs after the legitimate `ys` command, the desired value for `ys` is overwritten and the user is in for an interesting time. The HADES parser is case sensitive, which helps to reduce some of the potential for confusion. Because of this unpredictability, HADES stops and notifies the user in the event of embedded keywords.

Keywords that can only occur within a limited scope are called block commands. All block commands should be specified with the first line being the block specifier alone and the last line the block termination statement alone. Between these, only commands for the specified block are read in; lines specifying the position of the source in a block would be ignored. There is no limit to the number of these blocks in an input deck.

Array type variables must be alone on a line in the input deck. They may span more than one line, but no line with an array-type variable can have excess characters (except for comments, which are always allowed). Error checking is provided to make sure that the length of the array is not exceeded in the input deck.

String input can be provided with or without quotation marks. A string begins on the first non-white space character after the equal sign (if present) and ends on the first white space character after the string.

After HADES has been invoked and the input deck has been read in, HADES prints out to the terminal and log file any lines on which no keyword occurred. A line that says “BEGINNING

OF UNUSED LINE LISTING" signals these lines. A line that says "END OF UNUSED LINE LISTING" signals the end of this printout. The user should check this listing for mistyped commands (those with spelling or capitalization errors).

Syntactical Conventions

In what follows each HADES keyword will be summarized. At start of each entry, the keyword will be given at the start of the line, in `Typist` font. Tabbed 5 cm to the right is the explanation of the keyword. The first part of this explanation, in brackets, is a description of the expected argument(s) for the keyword. If the items in the bracket are in `Typist` font, exactly one of these options is expected. If the bracketed item is in *italics*, this is a description of what type of argument is expected.

General Body Commands

Positioning:

All directions oriented as if looking from source. All positions are relative to the center of the mesh to be radiographed. Note that the `camera` block, discussed on page 37, is an easier way to specify source and detector locations.

`xs`, `ys`, `zs` — [*number*] position of source in cm

`x1`, `y1`, `z1` — [*number*] upper-right hand corner of screen, in cm

`x2`, `y2`, `z2` — [*number*] lower-right hand corner of screen, in cm

`x3`, `y3`, `z3` — [*number*] lower-left hand corner of screen, in cm

`ifmax` — [*integer*] number of pixels in x direction on screen. For `DEF` (`CT`) `ORIENTATION`, this is the vertical (horizontal) direction on the screen.

`jfmax` — [*integer*] number of pixels in z direction on screen. For `DEF` (`CT`) `ORIENTATION`, this is the horizontal (vertical) direction on the screen.

`xcval` — [*integer*] This is a debugging command. Instead of computing all rays in the x direction ranging from 0 to ifmax-1, HADES will only compute the ray with value *xcval* in the x direction. This allows the code developer to concentrate on a particular ray for deeper study. This command only works for solid body objects and meshes which are inside the main ray bundle loops. Meshes such as `3Dhex` and `2DALExy` do not use `xcval`. This command must be used with `zcval`.

`zcval` — [*integer*] This is a debugging command. Instead of computing all rays in the z direction ranging from 0 to jfmax-1, HADES will only compute the ray with value *zcval* in the z direction. This allows the code developer to concentrate on a particular ray for deeper study. This command only works for solid body objects and meshes which are inside the main ray-bundle loops. Meshes such as

	3Dhex and 2DALExy do not use `zcval`. This command must be used with `xcval`.
`ORIENTATION`	[`DEF, CT`] This command specifies how the screen should be oriented. `DEF` is the default, or original orientation, and `CT` is a computed tomography orientation.
`LOCAL ANGLE TYPE`	[`ORIGINAL, CTX, CTY`] This command specifies which set of angles will be used for doing the local rotation of the mesh.
`theta`	[*number*] rotate the mesh by angle theta, in degrees, around x(z, z) axis for `ORIGINAL` (`CTX, CTY`) angle set.
`phi`	[*number*] rotate the mesh by angle phi, in degrees, around z(x, -y) axis for `ORIGINAL` (`CTX, CTY`) angle set.
`mesh center X`	[*number*] *X* position, in mesh coordinates, about which the mesh will rotate. Currently only implemented for `3Dhex`, `3Dfem`, and `3Dicf` meshes.
`mesh center Y`	[*number*] *Y* position, in mesh coordinates, about which the mesh will rotate. Currently only implemented for `3Dhex`, `3Dfem`, and `3Dicf` meshes.
`mesh center Z`	[*number*] *Z* position, in mesh coordinates, about which the mesh will rotate. Currently only implemented for `3Dhex`, `3Dfem`, and `3Dicf` meshes.

File Commands

Note that specifying a filename for an output file type (*i.e.* `outtif myout` for a TIFF file) automatically turns on the corresponding switch (`tiff on`) unless explicitly turned off (`tiff off`).

`viewimage`	[`on/off`] turn on the production of a View file. Default is `on`.
`outview`	[*filename*] name of View file to be written. Note that HADES automatically adds the extensions for all `out`*X* commands; if `outview` is specified as xray.sdt, the output files will be named xray.sdt.sdt and xray.sdt.spr. The default name for all `out`*X* commands is **xray** (**pray** if protons are specified as the probe).
`swap`	[`on/off`] byte swap the View file. Default is `off` except when running on a Compaq alpha.
`siloimage`	[`on/off`] turn on the production of a SILO file. Default is `on`.
`outsilo`	[*filename*] name of SILO file to be written. Now, a .silo extension is added.
`logfile`	[*filename*] name of the logfile to write. The logfile name is outxray.log during the calculation. After the calculation is complete, the default is outxray. The .log extension is added to the specified filename.

`back file`	[*filename*] name of back projection input file. This option is only used with HADES-CCG and is not fully developed.
`ccg`	[`on/off`] turn on coupling to the CCG optimizer for tomographic reconstruction. Default is `off`. This option is not fully developed.
`tiff`	[`on/off`] turn on the production of a radiograph using TIFF format. Default is `off`.
`outtif`	[*filename*] name of the TIFF file. The automatically added extension is `.tif`.
`revtif`	[`on/off`] reverse the colormap for a TIFF file; that is, assign the maximum value in the image a grayscale value of 0 and the minimum value in the image a grayscale value of 255. This operation can also be done in postprocessing, but would be tedious in the case of generating a large series of radiographs for a movie. Default is `off`.
`mintif`	[*number*] stretch color map of TIFF file down to this value, in units of pathlength. Default is to use the min of the image. Must be used in conjunction with `maxtif`.
`maxtif`	[*number*] stretch color map of TIFF file up to this value, in units of pathlength. Default is to use the max of the image. Must be used in conjunction with `mintif`.
`dump`	[`on/off`] turn on the production of a restart file. Default is `off`.
`outres`	[*filename*] name of the restart file. The automatically added extension is `.res`. This is a SILO file.
`pov`	[`on/off`] turn on the production of a POV-Ray visualization file. Default is `off`.
`outvis`	[filename] name of the visualization file. The automatically added extension is `.pov`. This is an ascii text file.
`py`	[`on/off`] turn on the production of a Python script file used to invoke charon on the SDT file. Default is `off`.
`outpy`	[filename] name of the Python script file. The automatically added extension is `.py`.
`outhist`	[*filename*] name of the file which contains track information about each ray. Only works for `object type 2Dxy` or `3Dxyz`. Default is `off`.
`nbin`	[*integer*] number of bins to use in histogramming each track. Only works for `object type 2Dxy` or `3Dxyz`.
`minhist`	[*number*] let lowest bin of track histogram start with this value. Only works for `object type 2Dxy` or `3Dxyz`.

`maxhist` [*number*] let highest bin of track histogram end with this value. Only works for `object type 2Dxy` or `3Dxyz`.

Control Commands

`nplot` [*integer*] For SILO input files with more than one mesh, specifies the number of plots to make

`skipcyc` [*integer*] For SILO input files with more than one mesh, specifies the number of meshes to skip before starting radiography.

`noobject` [`on/off/atall`] Turn on or off the capability of producing a file without the mesh. `on` creates radiograph with mesh, `atall` creates the radiograph without the mesh, and off creates the file with the mesh. Default is `on`. If the `atall` option is chosen, the code ignores the mesh name on the execute line. Thus, if only non-mesh objects are to be used in this radiograph, any junk can be put into the second argument of the command line. Note also that running HADES with no second argument is equivalent to running with `noobject` set to `atall` (i.e. only non-mesh objects are included in the radiograph). This command is essentially out of date, since it's effects can be achieved without using it.

`verbose` [`on/off`] Generate verbose output, sending messages not only to the logfile, but also to the standard output. Default is off. This keyword takes effect at its point of insertion in the input deck; verbose on at the end of the deck will forgo any verbose output during the read in of the deck.

`diagnostics` [`on/off`] Generate extra diagnostic output for code developer. Default is off. If this is on, extra images are produced. If an ALE mesh is being radiographed, the raw pathlength image is saved in a file with "prepost" appended onto the root name. If statistics or blurring are invoked for X-rays, an "n" is appended to the root name for the file containing the unperturbed pathlength image. If statistics and blurring are invoked for X-rays, an "nn" is appended to the root name for the file containing the unblurred image with statistical fluctuations.

`preview` [`on/off]` Use the *xv* imaging code to throw a preview image onto your screen automatically. This is useful when a quick check of the field of view is needed.

Mesh Manipulation Commands

`object type` [`1DALESPHERE`, `1DrASCII` , `2D`, `2DALErz`, `2DALErzMB` `2DALExy`, `2Dxy`, `3Dcart`, `3Dhex`, `3Dfem`, `3Dicf`, `3Dview1`, `3Dviewn`, `3Dxyz`] type of the mesh used. `1DALESPHERE` refers to a 1D radial ALE mesh. `1DrASCII` refers to a 1D radial mesh specified in an ASCII file. `2D` refers to a regular Cartesian r-z mesh (not an ALE mesh). `2DALErz` refers to a 2D r-z mesh that has several blocks, and is not Cartesian (an

	ALE mesh in r and z). `2DALErzMB` refers to a 2D r-z mesh that is not Cartesian (an ALE mesh in r and z) but is broken into several sub-meshes. `2DALExy` refers to a 2D x-y mesh that is not Cartesian (an ALE mesh in x and y). `2Dxy` refers to a 2D Cartesian mesh. `3Dcart` and `3Dxyz` refers to a 3D regular Cartesian mesh. `3Dhex, 3Dicf,` and `3Dfem` refer to 3D hexahedral meshes in various formats. `3Dview1` refer to a 3D Cartesian mesh in raw binary format which is comprised of a single material. `3Dviewn` is a 3D Cartesian mesh in raw binary format which is comprised of several materials.
`naminp`	[*filename*] name of file containing the isotopic composition of each material in the mesh in `isofrac()` form. See page 38 for more information on the use of the `isofrac` command. This option is required whenever `gamma energy`, `dn`, or `spectrumfile` are used.
`mac`	[*number number* …] give a list of mass absorption coefficients to be used, one for each material in the same order as the input file; units are cm^2/g. This option is no longer supported.
`flip`	[`on/off`] if on, reflect mesh about the X axis. This option is designed for 90^o 2D meshes. The default is off.
`refx`	[`on/off`] if on, reflect a 3D mesh about the X axis. Only works when `object type 3Dhex`, `3Dfem`, or `3Dicf` is set.
`refy`	[`on/off`] if on, reflect a 3D mesh about the Y axis. Only works when `object type 3Dhex`, `3Dfem`, or `3Dicf` is set.
`refz`	[`on/off`] if on, reflect a 3D mesh about the Z axis. Only works when `object type 3Dhex`, `3Dfem`, or `3Dicf` is set.
`2dALExy_ztop`	[*number*] specify the maximum z value to which the x-y mesh will be stretched. Required when `object type 2DALExy` is specified. It is preferred to use `ztop`.
`2dALExy_zbottom`	[*number*] specify the minimum z value to which the x-y mesh will be stretched. . Required when `object type 2DALExy` is specified. It is preferred to use `zbottom`.
`ztop`	[*number*] specify the maximum z value to which the x-y mesh will be stretched. Required when `object type 2DALExy` or `2Dxy` is specified.
`zbottom`	[*number*] specify the minimum z value to which the x-y mesh will be stretched. Required when `object type 2DALExy` or `2Dxy is` specified.
`.material multiply`	[*i value*] Multiply all densities of mesh material i by value. This option can be used to zero out undesirable materials. Note that the index *i* is 0-indexed. That is, for mesh materials, *i* ranges from 0 to number of materials – 1.

`keepinside` [*number*] specify the radius within which the mesh will be ray traced. All zones outside this radius will be ignored.

`keepoutside` [*number*] specify the radius outside which the mesh will be ray traced. All zones inside this radius will be ignored.

Source and Detector Commands

`Parallel Beam` [`on, off`] specify a ray-trace of the object using parallel beams instead of a cone beam. The detector pixels are used to specify the location of the rays such that they are all parallel. This option will only work when the *Y* axis is the radiographic axis.

`Neutron Radiography` [`on, off`] Choose neutrons as the probe for doing radiography. When this flag is `on`, all spectra are interpreted as neutron spectra and the ENDL tables are interpolated in order to compute attenuation coefficients for all isotopes specified.

`gamma energy` [*number*] specify a single xray energy for the shot. The EPDL tables are interpolated in order to compute attenuation coefficients for all isotopes specified. Units are MeV and final mass absorption coefficients are printed out to the logfile.

`neutron energy` [*number*] specify a single neutron energy for the shot. The ENDL tables are interpolated in order to compute attenuation coefficients for all isotopes specified. Units are MeV and final mass absorption coefficients are printed out to the logfile. This is one way of telling HADES that neutron radiography is desired.

`ienb` OPTION SUPERCEDED. NO LONGER SUPPORTED.

`dn` [*number number …*] Specify the dose or number in each energy bin. Values of `dn` need not be normalized.

`idflwt` OPTION SUPERCEDED. NO LONGER SUPPORTED.

`spectrumfile` [*filename*] Specify a file name for the spectrum and detector response functions. This file has four columns; each row corresponding to an energy bin. The first (second) column is the lower (higher) energy limit of the bin (in MeV). The third column is the Dose (in R at one meter) within this bin. The last column is the detector response within this bin and is only used in the third photon/neutron option. This number is a multiplier for the number of photons determined to be within this bin before statistical fluctuations are computed. Thus, a value of 1.00 has does not affect the computed dose for this energy bin, while a value of 2 doubles the number of photons hitting the pixel for this energy bin. The fourth column is ignored if the user has specified a detector DQE data file.

`number spectrum` [`on/off`] Tell the code whether the spectrum file is in units of photon flux (units of #/cm^2 at one meter). Default is off, meaning that a Dose spectrum is being given in the spectrum file. This

option only operates when the `spectrumfile` option has been implemented.

`per energy` [`on/off`] Tell the code whether the spectrum has an additional factor of 1/bin width in the spectrum. Spectra are best plotted in this format and this option allows such spectra to be directly plugged into the code, without user conversions. The default is `off`.

`invsqr` [`on/off`] Turn on or off inverse square weakening of the rays. Without any objects, the ray from the source to the center of the screen is assigned 1.0, points on the screen closer to or further from the center are assigned values corresponding to the relative inverse square weakening. Default is `off`. This option is not necessary when `Dose` has been specified; the inverse square weakening is already included in the dose conversions.

`SPOT type` [`GAUSSIAN,BENNETT,TOPHAT`] Specify a particular shape for the source spot size. This shape is then magnified appropriately and used for convolution at the detector plane. Currently, this is not used in the `spot` command for full convolution through the object.

`FWHM spot` [*number*] Specify the FWHM of the source spot of the X-ray or neutron source in units of cm. If the user does not specify a source spot type, a Gaussian is assumed. If the user has specified a `TOPHAT` spot profile, then the FWHM is interpreted to be the spot diameter. If `SPOT type` is `BENNETT`, the FWHM is the FWHM of the Bennett function.

`FWHM film` [*number*] Specify the FWHM of the blur at the detector plane due to the X-ray or neutron detector in units of cm.

`dqe data file` [*file*] Specify an input file that contains the energy-dependent response for the detector. See Detector Definition on page 53 for more information.

`detector blur file` [*file*] Specify an input file that contains the energy-dependent blurring for the detector.

`Image padding` [`ZEROPAD,UNITYPAD,EXTRAPOLATEPAD`] This option controls how the convolutions are done at the detector plane. When convolution at the detector plane is necessary, HADES performs an FFT of the image plane using an image size that is padded appropriately given the sizes of both the image and the blur function. This then leaves the question of what to do with pixels that are outside the image but in the extended image. The default is `ZEROPAD`, which surrounds the image with zeroes (typically, this corresponds to a collimated beam). In some cases it is more appropriate to use `UNITYPAD`, which surrounds the image with ones (typically, this corresponds to a small object within an extended field of illumination). In other cases, the user may prefer `EXTRAPOLATEPAD`, which takes the outermost values of the image

and continues these outward radially until the image is filled (typically, for calculating a lineout through a translationally symmetric object). See Detector Definition on page 53 for more information.

`Dose` [*number*] Specify the incident X-ray dose in units of Roentgen at 1 meter. This option cannot be used for `mac`. It may be used with `gamma energy`, and `dn`. It must be used when `spectrumfile` is invoked.

`XRay intensity` [*number*] Total number of X-ray photons that would impinge the field of view of the detector in the absence of attenuation. The resulting radiograph will have the statistical fluctuations for each pixel assuming Poisson statistics. This option may be used with `mac`, `gamma energy`, and `dn`.

`Xseed` [*integer*] A negative integer seed value for the random number generator used in generating Poisson fluctuations for X-rays. Default value is -11.

`spectrum at` [*zvalue xvalue*] specify a ray along which you wish to compute a dose spectrum in units of Roentgen/MeV. For a given energy bin, the product of the spectrum at this bin and the bin width in MeV gives the exposure in Roentgen for a pixel centered on the ray, where the pixel size is as was specified by the user. Detector sensitivity is not included in this exposure. The ray is defined as starting at the source center and running through the point at y=0 which is specified by *zvalue* and *xvalue* and which intersects the detector plane. The output is put in the file “spectrum.dat”. Each output spectrum has four columns. The first three columns are the lower edge, center, and upper edge, respectively, of the energy bin. The fourth column gives the exposure in this bin. This command can be invoked more than once. If it is, all of the detector spectra are put into “spectrum.dat” in the order in which the rays were specified. At the head of each spectrum is a line identifying the ray. This command only works with the `spectrumfile` method of specifying a source, and uses the spectrum specified there. At present, this option is not implemented for 3D meshes.

Background/Scattering Commands

While HADES is a ray-tracing code, and as such cannot estimate the amount of scattered background in an image, Monte Carlo calculations can provide such estimates. Currently, the user can specify a constant-height background or a radially-varying background. This background may also be energy-dependent. The keywords are:

`background_level` [*number*] specifies the height of the background relative to the signal height at the center of the image. By default, the background level is in terms of photon fluence. If the user

specifies a background energy spectrum, the background level is in terms of energy fluence.

`background_energy_spectrum_file` [*filename*] gives the filename of the ASCII file containing (x,y) pairs describing the energy spectrum of the background. The (x,y) pairs may be free-form. No header information is allowed within the file. If a background energy spectrum file is given, then the background level must also be given.

`background_radial_level_file` [*filename*] gives the filename of the ASCII file containing (x,y) pairs describing the radial dependence of the background. These values may be relative values, and will be scaled using the `background_level` value. The (x,y) pairs may be free-form and unevenly spaced. If this file is specified, then the `background_level` must also be given. The range of radii in the file must cover the radial range of the image.

Block Commands

Block commands are families of commands whose scope is limited to the to block in which they are defined. Blocks are used to define specialized radiographic probes (such as protons or X-Rays with a finite spot size) and to define solid body objects. Here is a list of the blocks currently available for use:

`protons`	specify protons as the radiographic probe.
`camera`	specify the detector location using the source, magnification and desired field of view.
`spot`	specify a finite sized x-ray source.
`multiangle`	specify phi, theta pairs for mesh rotation.
`plate`	specify a rectangular plate.
`sphere`	specify a sphere.
`ellipsoid`	specify an ellipsoid.
`cone`	specify a cone.
`collimator`	specify a graded collimator.
`gencoll`	specify a collimator as an ordered series of arcs and line segments.
`zrprofile`	specify an axially symmetric object whose outer boundary is defined by a set of (z,r) points.
`torus`	specify a torus.
`excluded object`	specify an excluded object.

Probe Blocks

Proton Block Commands

The proton commands choose protons as the radiographic probe for the simulation. If protons are specified, neither mac, dn, gamma energy, FWHM film, FWHM spot, nor invsqr can be specified since all of these parameters presuppose photons. If protons are specified, the isotopic compositions are needed, just as they are for monochromatic or multigroup X-rays. The user must use the naminp option (outside this block) to specify the isotopic composition. Most of the keywords described below are of interest to proton radiography hobbyists. The general user will need only the "cut" keyword (used twice) and either the "kenergy" keyword or the "prad beamline data file" keyword.

`protons`	begin block
`endprotons`	end block
`kenergy`	[*number*] specifies the incident kinetic energy of the protons, in units of GeV. A REQUIRED keyword, if the user has not specified a particular proton radiography beamline. If the user has specified a particular beamline, this key word is ignored unless the user also specifies "autotune off" (see below). In this case, kenergy is used to specify the energy to which the imaging magnets will be tuned. The incident proton beam energy is taken from the beamline data file.
`lateral blur`	[`on/off`] controls whether the code performs the final lateral blurring of the image. Default is `off`.
`log term`	[`on/off`] controls whether the code includes the logarithmic term in the expression for θ_o. Default is `off, (i.e., include the log term).`
`Nuclear Only`	[`on/off`] controls whether only the nuclear attenuation of the object will be computed. This option was developed for tomography applications. Default is off.
`RadLen Only`	[`on/off`] controls whether only the radiation length of the object will be computed. This option was developed for tomography applications. Default is `off`.
`cut`	[*number*] specify the angle (in milliRadians) used for the first and second cut. A positive value indicates that a collimator with a maximum specified angle is desired. A negative value indicates that an anticollimator with a minimum specified angle is desired. HADES assumes that the user in interested in two angular cuts, so therefore this keyword must appear twice within the proton block. This is a REQUIRED keyword.
`pdata`	[`ppscale,letaw`] specify the form of proton data to use. `ppscale` interpolates the p-p data from the Particle Data Group and then scales it for A. `letaw` uses the fit of Hartouni for Hydrogen isotopes if E<10 GeV, (PDG data otherwise) and the

Letaw phenomenological fits for all other nuclei. For radiation lengths, the formula of Dahl is used. The default is `letaw` (this is much better). See Appendix G: Data Used and Locations for references to Dahl and Letaw.

`intensity` [*number*] Total number of protons which would impinge the field of view of the detector in the absence of attenuation. The resulting radiograph will have the statistical fluctuations for each pixel assuming Gaussian statistics.

`seed` [*integer*] A negative integer seed value for the random number generator used in generating Gaussian fluctuations. Default value is -11.

`FWHM` [*number*] a value in cm, giving the FWHM of the Gaussian intensity distribution of the protons at the object plane. This option is useful for studying noise effects at the edge of an image. If no value is given, a flat distribution is assumed.

`prad beamline data file` [*filename*] lets the user specify a file containing the apertures, lengths, strengths, and spacings of the magnets in a proton radiography beamline. The data file also contains the nominal beam energy for the given beamline, the nominal FWHM of the intensity distribution at the object plane, and the lengths of the angle-selecting collimators.

`autotune` [`on/off`] in combination with a user-specified prad beamline file, controls whether HADES will adjust the magnet settings to correct for dE/dx changes in the beam energy after the protons have passed through the object. Default is ON. If this keyword is set to OFF and the user has not specified a desired tune energy via the kenergy command, HADES will use the nominal beam energy from the prad data file.

`nbeamlets` [`integer`] in combination with a user-specified prad beamline file, lets the user control (typically, decrease) the number of beamlets used to calculate the blurring effects of mistuned magnets. The default value is 10201, which appears to give visually credible estimates of limbing, etc. If the user is comfortable with noisier images (which may be adequate for estimating changes in overall image intensity), nbeamlets may be decreased.

All proton commands must be within the `protons`/`endprotons` block to be valid.

Spot Block Commands

The spot commands allow the user to specify a finite sized x-ray spot and to do a full convolution through the object, rather than just a convolution at the detector plane. This spot is assumed to have an axial symmetry. HADES samples the source using an r-ϕ mesh centered at the source location specified in the `xs, ys, zs` commands. The user can specify the angular sampling, which spans 0^o to 360^o. The user specifies the radial sampling

by defining the intensity vs. r profile. The resulting r-ϕ mesh will typically require many ray-tracing calculations. HADES can do these calculations in parallel, since this kind of simulation is computationally intensive.

`spot`	begin block.
`endspot`	end block.
`Number of Phi Zones`	[integer] specify the number of angular zones to use in dividing the 360^o range. The default value is 36 (10^o increments).
`pairs`	This begins the sub-block which defines the (r,intensity) pairs in the profile. HADES expects one (r,intensity) pair on each successive line until the sub-block is ended. On each line, the r value should be first, followed by the intensity value.
`endpairs`	end the pairs sub-block.

All spot commands must be within the `spot`/`endspot` block to be valid.

Solid Body Blocks

Commands Used in All Solid Body Blocks

All of these blocks are used to specify solid bodies to add to the problem. There is no absolute upper limit to the number of solid bodies that the user can specify with HADES. The following comments apply to solid bodies in general.

Currently, HADES has no overlap checking algorithms. This user can use this feature to sculpt features inside a solid body by adding other objects inside with negative density. For example, a spherical shell can be built by specifying a sphere of positive density and nesting another sphere inside it, with negative density of the same magnitude. Of course, this approach requires that the user be very careful that the setup is properly specified. HADES also has constructive solid geometry operations such as intersection, union and difference. This capability is discussed in the section on excluded objects on page 33. New options allow the user to obtain a rendering of the radiographic setup before running the code.

All solid body blocks use some commands in the same way. In order to prevent unnecessary repetition, we will define these common commands in this section. First we will discuss commands which rotate the object in some way. As HADES has developed, we have found that it is useful to define two sets of rotation angles. The first set of rotations is called the local set and is used to orient the object about its center. The second set of rotations is called the global set and can be used to orient the object about a user-defined center. All objects first undergo their local rotation, followed by their global rotation. The global rotation is useful for radiographing collections of objects from different angles, as is done in computed tomography. Each set of rotations can be defined using unique systems of angles: `ORIGINAL`, `CTX` and `CTY`, as defined above on page 13.

`LOCAL ANGLE TYPE`	[`ORIGINAL, CT`] This command specifies which set of angles will be used for doing the local rotation.

`theta` [*number*] local angle, in degrees, through which the object is to be rotated about the first rotation axis.

`phi` [*number*] local angle, in degrees, through which the object is to be rotated about the second rotation axis.

`GLOBAL ANGLE TYPE` [`ORIGINAL, CT`] This command specifies which set of angles will be used for doing the global rotation.

`GLOBAL X,Y,Z` [*number*] specifies the center for the global rotation of the object.

`GLOBAL THETA` [*number*] global angle, in degrees, through which the object is to be rotated about the first rotation axis.

`GLOBAL PHI` [*number*] global angle, in degrees, through which the object is to be rotated about the second rotation axis.

`MultiProjection` [`on/off`] if `on`, object is rotated as specified by a rotation block (see `Mesh Rotation block`). Default is `off`.

The second class of commands common to all solid body blocks involves the specification of material properties. The `isofrac` and `rho` commands are required for all solid-body objects.

`isofrac` [*integer number* …] specifies an ordered list of isotopes specifying the composition of the object. See notes on isofrac lines on pages 38 and following.

`rho` [*number*] specifies the density of the object in g/cm^3.

`mac` [*number*] specifies an x-ray attenuation coefficient for the object in cm^2/g. This option cannot be used unless it is used for all solid body objects and the mesh. This option precludes the used of the `gamma energy` or `spectrum file` options. Because our treatment of photons and neutrons is quite sophisticated, this option is almost obsolete.

`MUp` [*number*] specifies the nuclear attenuation coefficient for the object in units of cm^2/g. Only relevant when protons are the radiographic probe. This allows the user to use a value obtained by fitting step wedges.

`X0` [*number*] specifies the radiation length for the object in units of g/ cm^2. Only relevant when protons are the radiographic probe. This allows the user to use a value obtained by fitting step wedges.

`DEDX` [*number*] specifies the energy loss of protons through the object in units of MeV cm^2/g. Only relevant when protons are the radiographic probe. This allows the user to use a value obtained by fitting step wedges.

`label` [*string*] specifies a text string that is used to identify the block in the output file.

Plate Block Commands

The plate commands allow rectangular prisms to be added to a setup. Any rectangular prism may be added, but they must first be specified as perpendicular to the coordinate system.

`plate` begin block

`endplate` end block

`xmin,ymin,zmin` [*number*] specifies the lowest x, y, and z values for the rectangular prism

`xmax,ymax,zmax` [*number*] specifies the highest x, y, and z values for the rectangular prism

`xlength, ylength, zlength` [*number*]

specifies the length of the rectangular prism in the x, y, and z directions, respectively. If any of these parameters are set, all of them must be set, in addition to the centering parameters listed on the next line. If these parameters are set, the `x,y,zmin,max` parameters listed above are overwritten so as to be consistent with these specifications.

`xcenter, ycenter, zcenter` [*number*]

specifies the x, y, and z positions of the rectangular prism's center. These parameters can only be used in conjunction with the `x,y,zlength` parameters just described.

All plate commands must be within the `plate`/`endplate` block to be valid.

Sphere Block Commands

The sphere commands allow spheres to be added to a setup. A spherical cap can be specified by using the `yslice` command.

`sphere` begin block

`endsphere` end block

`x,y,z` [*number*] specifies the x, y, and z values for the sphere's center.

`yslice` [*number*] specify a spherical cap with the base formed by the y = *yslice* cut plane. The cap is taken to be all parts of the sphere for which y is greater than this cut plane. Subsequent rotation can reorient this object, but note that `theta` and `phi` rotate relative to the full sphere's center.

`radius` [*number*] the sphere's radius.

All sphere commands must be within the `sphere`/`endsphere` block to be valid.

Ellipsoid Block Commands

The ellipsoid commands allow ellipsoids to be added to a setup. An elliptical cap can be specified by using the `yslice` command.

`ellipsoid`	begin block
`endellipsoid`	end block
`x,y,z`	[*number*] specifies the x, y, and z values for the ellipsoid's center.
`A,B,C`	[*number*] specifies the length of the x, y, z axis for the ellipsoid.
`yslice`	[*number*] specify a ellipsoidal cap with the base formed by the y = `slice` cut plane. The cap is taken to be all parts of the ellipsoid for which y is greater than this cut plane. Subsequent rotation can reorient this object, but note that `theta` and `phi` rotate relative to the full ellipsoid's center.

All ellipsoid commands must be within the `ellipsoid`/`endellipsoid` block to be valid.

Cone Block Commands

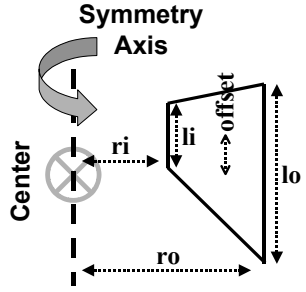
The cone commands allow right circular cones to be added to a setup. The specification is such that the altitude is parallel to the y axis. Rotations will allow other orientations.

`cone`	begin block
`endcone`	end block
`x, y, z`	[*number*] specify the center of the cone.
`radius`	[*number*] radius of the base circle of the cone.
`height`	[*number*] height of the cone from tip to base.

All cone commands must be within the `cone/endcone` block to be valid.

Collimator Block Commands

The collimator commands allow graded collimators to be added to a setup. A graded collimator is a rectangular prism in r-y coordinates, rotated about the y axis. The figure at right shows the definition of the parameters discussed below.



`collimator`	begin block
`endcollimator`	end block
`x,y,z`	[*number*] the center of the collimator, in cm.

`ro`	[*number*] the outer radius of the collimator
`ri`	[*number*] the inner radius of the collimator
`li`	[*number*] the length of the inner cylinder of the collimator. This is a length in the y direction.
`lo`	[*number*] the length of the outer cylinder of the collimator. This is a length in the y direction.
`offset`	[*number*] the displacement of the inner cylinder's center along the y axis from the center of the collimator. The center of the collimator is the center of the outer cylinder. This displacement is in the same sense as the y coordinate axis.

All collimator commands must be within the `collimator/endcollimator` block to be valid. Note that `lo` and `li` individually can have values of zero, in which case the collimator tapers to a point.

General Collimator Block Commands

The general collimator commands allow the specification of collimators of odd shapes and sizes. Basically, the collimator is bounded by two y=constant planes. On the front, a shape made of arcs and line segments is specified. A similar shape of corresponding arcs and line segments is specified for the back. The two sides are then connected with a smooth connecting surface. The "front" side of the collimator is the side on the traditional source side, i.e. the side that is at a more negative Y value. A worked example of this object can be found in Example 6 of Appendix A: Example Input Decks on page 49.

`gencoll`	begin block
`endgencoll`	end block
`yfront`	[*number*] the y-coordinate (in cm) of the front of the collimator
`yback`	[*number*] the y-coordinate (in cm) of the back of the collimator
`arcf`	[*number number number*] The three numbers are x, y, and radius. This command specifies that an arc from a circle centered on (x,y) and with a given radius is a part of the boundary of the front of the collimator. SEE BELOW FOR THE COORDINATE SYSTEM BEING USED IN THIS COMMAND.
`segmentf`	[*number number number number*] The four numbers are x1, y1, x2, and y2. This command specifies that a segment of a line which goes through the points (x1,y1) and (x2,y2) is a part of the boundary of the front of the collimator. SEE BELOW FOR THE COORDINATE SYSTEM BEING USED IN THIS COMMAND
`arcb`	[*number number number*] Same as `arcf`, but for the back of collimator.

`segmentb` [*number number number number*] Same as `segmentf`, but for the back of collimator.

`flip` [`on/off`] Flip the given top-to-bottom order of the gencoll shape descriptors Turning on `flip` may invalidate a correctly specified `gencoll` . Default is `off`.

A NOTE ON COORDINATE SYSTEMS: the commands `arcf`, `segmentf`, `arcb`, and `segmentb` use a special coordinate system, just for the ease of specifying arcs and segments. On the front and back planes (specified by `yfront` and `yback`), a special x-y coordinate system is superimposed, so that the X-axis is horizontal and the Y-axis is vertical, when viewed from the source. In this way, the normal (x,y) specifications for curves can be worked out by the user and used in the simulation. Note that in HADES coordinates, the "+Y" axis at right is actually the +X axis, while the "+X" axis at right is actually the -Z axis.

The shapes of the front and the back of the collimator must be specified in a careful order. The order is used to determine which part of the full line or full circle to use (intersections with neighboring lines or circles determine the extent of the segment or arc). The topmost point (highest value of y) must be on the first specified piece of the shape, whether an arc or segment. Then, the pieces must be listed in order counterclockwise around the shape. Valid shapes must be convex, that is, the inner angle between two intersecting curves must be greater than 90^{o}. Also, no line/arc intersections where the line is perpendicular to the tangent of the circle at the point of intersection are permitted. Breaking either of these specifications will not result in an error; instead, an unpredictable shape will result.

In order to smoothly connect the front and the back of the collimator, it is required that they be "scaled" versions of each other. In other words,

(1) Each segment and/or arc on the front must have a corresponding piece of the same type on the back

(2) The segments and arcs on the back must be proportional in length to those on the front, and

(3) The arcs must subtend the same angle on the front and the back.

Violation of the first of these limitations will result in an error. Violations of the second and third limitations are allowed, but the results are once again not strictly predictable.

ZRProfile Block Commands

A zrprofile is an axially symmetric body enclosed by a perimeter of (z,r) pairs. The symmetry axis is taken to be the Z axis, so that the object will be radiographed in profile, unless the user rotates it. If the first and last points specified in the profile are at r=0, HADES does not try to connect the first point to the last point. The perimeter is assumed to start and end on the Z axis. Otherwise, the segment connecting the first and last points is included in the perimeter.

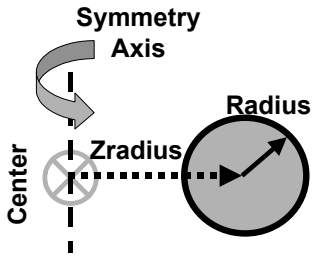
`zrprofile` begin zrprofile block.

`endzrprofile` end zrprofile block.

`x,y,z`	[*number*] The center of the object.
`pairs`	This begins the sub-block which defines the (z,r) pairs in the perimeter. HADES expects one (z,r) pair on each successive line until the sub-block is ended. On each line, the z value should be first, followed by the r value.
`endpairs`	end the (z,r) pair definition sub-block.
`eps`	[*number*] The fractional tolerance for intersection comparison. The default value is 10^{-4}. This value is typically good enough, but for perimeters defined by many points, a smaller value may be necessary.

Torus Block Commands

The torus command allows the user to add a torus to the setup. The user specifies the center of the object, the radial thickness of the torus, and the radius of the toroidal ring. HADES models this object internally as a `zrprofile`. The image at right shows the defining parameters of the torus.



`torus`	begin torus block.
`endtorus`	end torus block.
`x,y,z`	[*number*] The center of the object.
`Zradius`	[*number*] specify the radius of the toroidal ring.
`Radius`	[*number*] specify the radial thickness of the torus.
`number of pairs`	[*integer*] specify the number of pairs to be used in the ZRProfile. The default is 88.
`eps`	[*number*] The fractional tolerance for intersection comparison. The default value is 10^{-4}. This value is typically good enough, but for perimeters defined by many points, a smaller value may be necessary.

Excluded Object Block Commands

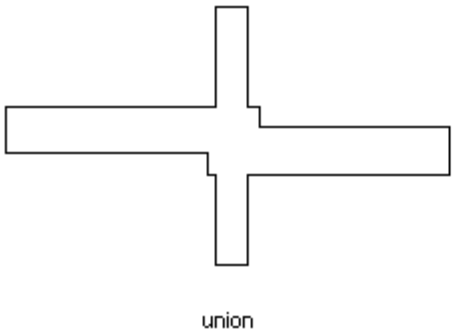
The excluded object commands allow the user to define a solid body using constructive solid geometry operations such as union, intersection, and difference. Excluded objects are compound objects built from two or more solid body objects. Excluded objects can be nested within other excluded objects. The excluded object is built by combining the solids using one of the Boolean operators: union, difference, or intersection. A fourth operation, ysum, can also be used. The operations are performed in a sequential fashion relative to the first specified solid in the excluded object; order is therefore very important.

`excluded object`	begin excluded object block.
`endexcluded object`	end excluded object block.

`xcenter,ycenter,zcenter`	[*number*] The center of the object.
`union`	specify a union Boolean operation
`difference`	specify a difference Boolean operation
`intersection`	specify a intersection Boolean operation
`ysum`	specify a ysum Boolean operation

Excluded objects must specify at least two solids, which may be any of the solids (except collimators) discussed above. Only the position and rotation data need be specified. The material data for each solid shouldn't be specified, since the excluded object itself requires the material properties. All of the constituent solids are treated together as a single object to be radiographed, so specifying `phi` and/or `theta` for the excluded object will cause each of the solids to be effectively rotated.

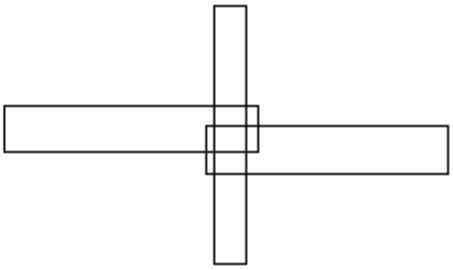
To give concrete examples of the four operators, three plates will be combined. The view is along the source-detector axis and the plates are aligned so that only one face for each is perpendicular to and visible to the axis (see below). The union operator creates an excluded



union

intersection

difference



ysum

object composed of the volume all three plates. The difference operator creates an excluded object composed of the volume of first plate, less the pieces that are also belong to either of the other two plates. The intersection operator creates an excluded object composed of the volume the first plate that also belongs to both of the other two plates. The ysum operator is similar to the union operator, except that the contributions of the common volumes from the

three plates are additive (as if the three plates were specified as individual objects, not as one excluded object), not inclusive.

All excluded object commands must be within an `excluded object`/`endexcluded object` block to be valid.

Mesh Rotation Blocks

The mesh rotation blocks define several ways to specify how to rotate a mesh through several angles, generating a separate radiograph at each angle. The `Multiangle Mode` command specifies which multiangle approach will be used. It applies to all mesh rotation blocks, and is external to the block. These rotations are also applied individually to solid body objects, if `MultiProjection` is set to `on` for each object. In this case the specified rotation angles are the global rotation angles about the global center defined for each object. There are five types of loops over rotation angle: `multiangle`, `multiphi`, `multitheta`, `multi_thetaphi`, and `sinogram`.

`Multiangle Mode` [*integer*] switch to turn on mesh rotation. Default is 0 which ignores all looping options. For `multiangle` and `multiphi`, the mode is 1. For `sinogram` and `multitheta`, the mode is 2. For `multi_thetaphi` with the inner loop over φ, the mode is 3. For `multi_thetaphi` with the inner loop over θ, the mode is 4.

Multiangle Commands

The multiangle command defines a set of `phi` and `theta` angle pairs for rotation. The angles are defined as above in the Positioning section for General body commands. HADES expects one (phi, theta) pair on each successive line until the sub-block is ended. On each line, the phi value should be first, followed by the theta value, without any separating comma. Set `Multiangle Mode` to 0.

`multiangle` begin multiangle block.

`endmultiangle` end multiangle block.

Multiphi Commands

The multiphi command defines a set of `phi` and `theta` angle pairs for mesh rotations. The angles are specified by a constant `theta` and a start, stop and number of `phi` angles. There will be (`phi_end`-`phi_start`)/`rot_num` rotations. . Set `Multiangle Mode` to 1.

`multiphi` begin multiphi block.

`endmultiphi` end multiphi block.

`theta` [*number*] local angle, in degrees, through which the object is to be rotated about the first rotation axis. Default is 0.

`phi_start` [*number*] local starting angle, in degrees, through which the object is to be rotated about the second rotation axis.

`phi_end`	[*number*] local ending angle, in degrees, through which the object is to be rotated about the second rotation axis.
`rot_num`	[*integer*] number of phi angles. There will be `rot_num` +1 separate radiographs generated, one per rotation angle. Default is 1.

Multitheta Commands

The multitheta command defines a set of `phi` and `theta` angle pairs for mesh rotations. The angles are specified by a constant `phi` and a start, stop and number of `theta` angles. There will be (`theta_end`-`theta_start`)/`rot_num` rotations. . Set `Multiangle Mode` to 2.

`multitheta`	begin multitheta block.
`endmultitheta`	end multitheta block.
`phi`	[*number*] local angle, in degrees, through which the object is to be rotated about the second rotation axis. Default is 0.
`theta_start`	[*number*] local starting angle, in degrees, through which the object is to be rotated about the first rotation axis.
`theta_end`	[*number*] local ending angle, in degrees, through which the object is to be rotated about the first rotation axis.
`rot_num`	[*integer*] number of phi angles. There will be `rot_num` +1 separate radiographs generated, one per rotation angle. Default is 1.

Multi_thetaphi Commands

The multi_thetaphi command defines a set of `phi` and `theta` angle pairs for mesh rotations. The angles are specified by a constant `phi` and a start, stop and number of `theta` angles, repeated over all phi angles. There will be (`theta_end`-`theta_start`)/`num_theta*` (`phi_end`-`phi_start`)/`num_phi` rotations. . Set `Multiangle Mode` to 3 (`phi` increments first) or 4 (`theta` increments first).

`multi_thetaphi`	begin multi_thetaphi block.
`endmulti_thetaphi`	end multi_thetaphi block.
`theta_start`	[*number*] local starting angle, in degrees, through which the object is to be rotated about the first rotation axis.
`theta_end`	[*number*] local ending angle, in degrees, through which the object is to be rotated about the first rotation axis.
`num_theta`	[*integer*] number of phi angles. There will be (`num_theta`+1)*(`num_phi`+1) separate radiographs generated, one per rotation angle/ Default is 1.
`phi_start`	[*number*] local starting angle, in degrees, through which the object is to be rotated about the second rotation axis.

`phi_end`	[*number*] local ending angle, in degrees, through which the object is to be rotated about the second rotation axis.
`num_phi`	[*integer*] number of phi angles. There will be (`num_theta`+1)*(`num_phi`+1) separate radiographs generated, one per rotation angle. Default is 1.

Sinogram commands

The sinogram command defines a set of `phi` and `theta` angle pairs for mesh rotations. The rotations are done with `phi` remaining constant, and `theta` changing as specified below. Unlike the mesh rotation blocks described above, only one radiograph is generated for the sinogram block. Each `theta` rotation adds one slice to the overall radiograph. Sinograms are always performed with `ORIENTATION` set to CT, `LOCAL ANGLE TYPE` set to CTX, and `Multiangle Mode` set to 2.

`sinogram`	begin `sinogram` block.
`endsinogram`	end `sinogram` block.
`start`	[*number*] local starting angle, in degrees, through which the object is to be rotated. Default is 0.
`stop`	[*number*] local ending angle, in degrees, through which the object is to be rotated. Default is 180.
`delta`	[*number*] local incremental angle, in degrees, through which the object is to be rotated.
`row`	[`on/off`] `on` to select a row slice, `off` to select a column slice. Default is `on`.
`index`	[*integer*] row (column) slice to radiograph. The slice number is relative to `ifmax` (`jfmax`).
`angles`	[*integer*] number of angles through which the object is to be rotated.

Miscellaneous Blocks

Camera Block Commands

The camera block allows the user to specify the detector in terms of the source, object center, desired magnification, and the field of view at the object. This option can greatly simplify the process of setting up a reasonable radiograph. It can also be useful for making radiographic movies of a fly-by of an object. The user specifies the source point and a point that will be treated as the center of the object. The user then species the desired field of view at the object center and the magnification of the object. These parameters are then used to compute the appropriate values of x1,2,3, y1,2,3, and z1,2,3.

`camera`	begin block
`xs`, `ys`, `zs`	[*number*] position of source in cm
`xo`, `yo`, `zo`	[*number*] position of center of object in cm

`MAG`	[*number*, greater than 1] the radiographic magnification of the system.
`FOV`	[*number*] the full-width of the field of view at the object plane, in cm.
`endcamera`	end block

Specification of Materials in a Problem

HADES must know the isotopic composition of all materials in a problem, in order to properly compute the attenuation coefficients for each material. Isotopic compositions are specified for materials using the `isofrac` command. The exact form of `isofrac` used varies depending upon whether mesh materials or solid-body object materials are being specified.

Meshes typically have several materials defined, and the `naminp` command defines the isotope file that has these definitions. Within this file will be a series of lines of the form:

`isofrac` (1) = …
`isofrac`(2) = …
.
.
.

Note that the numbering MUST start with 1 and proceed upward. There must be an `isofrac` for every material in the mesh. Comments are allowed in these files using the standard HADES comment specifier (//). These comments must either take a whole line, or begin after all the required parameters on an `isofrac` line. As always, HADES ignores everything on a line that follows a comment specifier.

For use on solid-body objects, `isofrac`() lines appear in the HADES input deck and have a format slightly different from those of isotope files. Unlike the mesh specification, there is only one material for each solid body, hence only one isofrac line needed in each solid-body. This material can be composed of several isotopes and the form of an isofrac line is

`isofrac` = …

where the only separators allowed between different numbers on the input line are spaces, and all isotopes must be put on the same line. The equal sign may be omitted, but NO OTHER CHARACTERS SHOULD APPEAR BETWEEN THE ISOFRAC KEYWORD AND ITS ARGUMENTS! The results are unpredictable, quite often resulting in the assignment of zero as the identifier of the object.

What follows the equal sign in both cases is a list of isotope names and their number or mass fractions: `isofrac` = $isotope_0\ frac_0\ isotope_1\ frac_1\ .\ .\ isotope_{n-1}\ frac_{n-1}$, where there are n isotopes in this material. The isotope name ($isotope_n$) for isotope ^{A}Z is Z*1000+A, so, for example, ^{56}Fe has the name 26056. To obtain the naturally occurring element, set A to be 0. Thus, naturally occurring iron is 26000. The fractions ($frac_n$) can be number fractions or mass fractions. If any of the fractions is negative, HADES assumes that mass fractions are in use. Otherwise, number fractions are assumed. HADES automatically normalizes the total fractions by forming a total sum and dividing each fraction by this sum. This behavior can be

useful for specifying a material with a known chemical formula. For example, if a material 10 were water, its specification would be:

`isofrac`(10) = 1001 2 8016 1

As a last example, consider a sphere made of an alloy which is 90% Fe by weight and 10% Co by weight:

`isofrac` = 26000 -0.9 27000 -0.1

Old-style Collimator Commands

These commands were the earliest ways of specifying a collimator in HADES. There are better, more general, block commands below. The user is encouraged to use them. An old-style collimator is converted to two graded collimator blocks. These commands only remain for backward compatibility.

`cfront`	[*number*] distance (in y coordinate) from front of collimator to source, in cm.
`cback`	[*number*] distance (in y coordinate) from back of collimator to source, in cm.
`rfront`	[*number*] the radius, in cm, of the front of the collimator
`rback`	[*number*] the radius, in cm, of the back of the collimator
`crho`	[*number*] the density of the collimator, in g/cm^3.
`matcol`	[*integer*] the number of the mesh material to use for this collimator. The materials are numbered in the same way they are for materials in the mesh itself.
`isofrac`	*[integer number* …] specifies and ordered list of isotopes specifying the composition of the material. If this option is used, `matcol` is not needed. See notes on `isofrac` below.

`catt` [*number*] the attenuation coefficient for the collimator. Either `matcol` or `catt` may be used, using both causes an error.

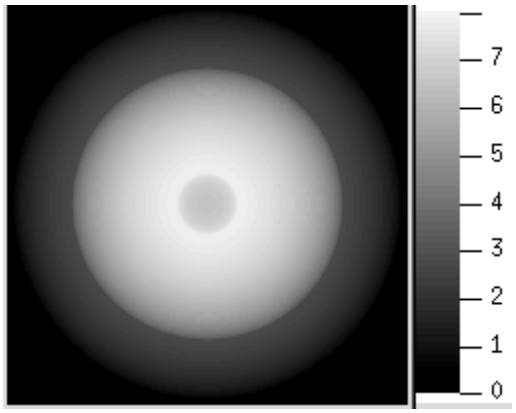
Appendix A: Example Input Decks

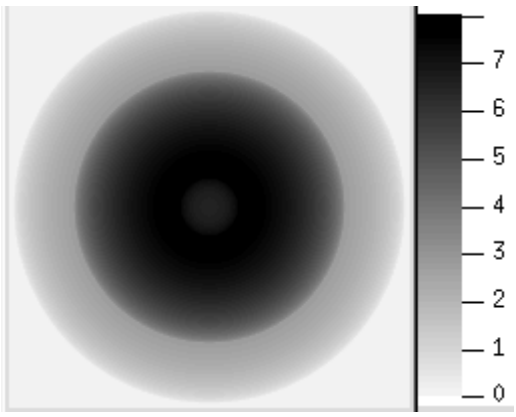
In this section, we have given six sample HADES problems, designed to capture as many features of the code as possible. For each example, we begin with a brief description of the simulation, followed by a listing of the HADES input deck used and also all subsidiary files used. A line of dashes delimits all subsidiary files: ----------. We also display the resulting radiographs as positives and negatives. The units (in pathlengths) are given using the color bar at the right of each image. It should be noted that these radiographs are grayscale images and are best viewed on a computer screen. When printed out, the appearance of the radiographs will vary with the quality of the printer used. We have found that only high-quality dye-sublimation printers are able to produce accurate hardcopies of these images. Poorer quality printers have artifacts in the images such as noisiness, or a poor stretch of the color map from white to black.

1. Nested spherical shells using solid body objects and no mesh

In this example, we radiograph two nested shells, of tungsten and copper, with 4 MeV photons. There is a 1 cm radius hole in the center of the shells. The radiographs are output in the files: xray.sdt (binary file), xray.spr (ASCII header), xray.pgm (Portable Greyscale Map image), and example1.tif (TIFF image). The shells are defined by using nested spheres of positive and negative density, to zero out the inner material.
Obtained by typing *hades example1.hades*

```
//Start of deck example1.hades
verbose on
preview on
gamma energy = 4
ifmax = 200
jfmax = 200
x1    = +20
y1    = 200
z1    = -20
x2    = -20
y2    = 200
z2    = -20
x3    = -20
y3    = 200
z3    = +20
xs    = 0.
ys    = -100
zs    = 0.
tiff on
outtiff example1
sphere // Shell of Copper
  x = 0.0; y = 0; z = 0.0
  radius = 6.5
  isofrac = 29000 1.0
  rho = 8.96
```





```
endsphere
sphere // Shell of Copper; erase inner stuff
  x = 0.0; y = 0; z = 0.0
  radius = 4.5
  isofrac = 29000 1.0
  rho = -8.96
endsphere
sphere // Shell of W
  x = 0.0; y = 0; z = 0.0
  radius = 4.5
  isofrac = 74000 1.0
  rho = 19.3
endsphere
sphere // Shell of W; erase inner stuff
  x = 0.0; y = 0; z = 0.0
  radius = 1
  isofrac = 74000 1.0
  rho = -19.3
endsphere
// End of deck
```

2. Nested spheres using excluded objects and detector parameters

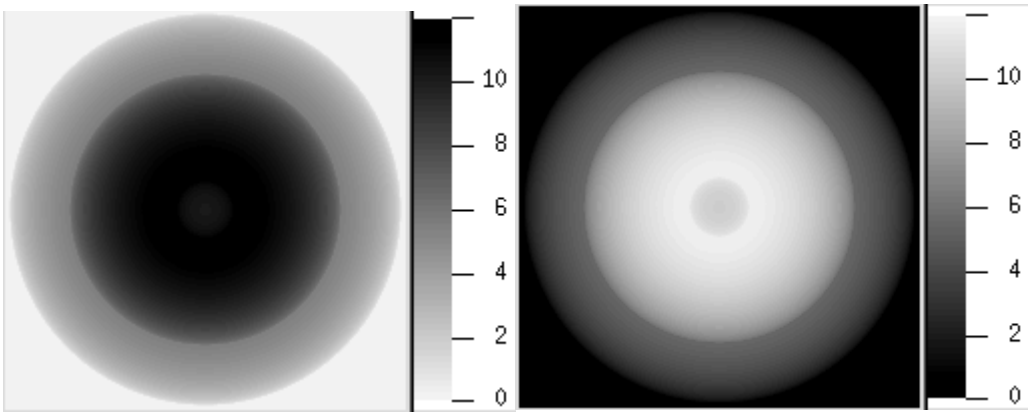
In this example, we simulate the same nested shells as in example 1, except that we define the shells using excluded objects instead of negative densities. Also, we run with an X-Ray spectrum and a set of detector files. All of the subsidiary files are listed below. The X-Ray spectrum is a 2 MeV endpoint bremsstrahlung spectrum generated from an MCNP[2] simulation of 2 MeV electrons colliding with a Tungsten target severel electron ranges thick. The effective energy of this source is much lower than the 4 MeV line source used in Example 1 and this can be seen in the longer pathlengths. The detector is a fragment of a full model for the energy deposit in a 2 cm thick slab of Bismuth Germanate. For brevity, we have pruned the detector blur model so that the blur function is modeled with 50 radial points at energies of 20 keV, 200 keV and 2 MeV. For the energy bins specified in the spectral file, HADES does a linear interpolation between the blur functions given in the blur file. Our energies are much too coarse, but this is meant to be a simple example. A dose of 5 Roentgens at 1 meter also has been specified. With a high-quality printout the statistical fluctuations can be seen.
Obtained by typing *hades example2.hades*

Input Deck and Radiograph

```
//Start of deck example2.hades
verbose on
preview on
spectrumfile 2MeVBremSpectrum_example2
detector blur file bgo_2cm_blur_example2.dat
dqe data file bgo_2cm_dqe_example2.dat
number spectrum on
```

[2] J.F. Briesmeister, "MCNP--A general Monte Carlo n-particle transport code", Technical Report LA-13709-M, Los Alamos National Laboratory, 2000.

```
Image padding ZEROPAD
Dose 5
ifmax = 200
jfmax = 200
x1    = +20
y1    = 200
z1    = -20
x2    = -20
y2    = 200
z2    = -20
x3    = -20
y3    = 200
z3    = +20
xs    = 0.
ys    = -100
zs    = 0.
tiff on
outtiff example2
outview example2
excluded object  //Shell of Copper
  xcenter = 0; ycenter = 0; zcenter = 0
  sphere // Shell of Copper
    x = 0.0; y = 0; z = 0.0
    radius = 6.5
  endsphere
  sphere
    x = 0.0; y = 0; z = 0.0
    radius = 4.5
  endsphere
  difference
  isofrac = 29000 1.0
  rho = 8.96
endexcluded object
excluded object  //Shell of Tungsten
  xcenter = 0; ycenter = 0; zcenter = 0
  sphere
    x = 0.0; y = 0; z = 0.0
    radius = 4.5
  endsphere
  sphere
    x = 0.0; y = 0; z = 0.0
    radius = 1
  endsphere
  difference
  isofrac = 74000 1.0
  rho = 19.3
endexcluded object
//End of deck
```



Spectrum File Used

This file is 2MeVBremSpectrum_example2. Note that the Tungsten K emission line is not well resolved. For some applications, this may be an unpardonable simplification. The actual file is between the "----" lines. This file is 46 lines long.

E_{low}	E_{high}	# $\gamma s/cm^2/e^-$	Not used, but expected
0.000000E+00	1.000000E-02	3.777480E-08	1.000000E+00
1.000000E-02	2.000000E-02	5.577390E-08	1.000000E+00
2.000000E-02	3.000000E-02	1.170060E-07	1.000000E+00
3.000000E-02	4.000000E-02	1.657460E-07	1.000000E+00
4.000000E-02	5.000000E-02	1.894420E-07	1.000000E+00
5.000000E-02	6.000000E-02	2.277090E-07	1.000000E+00
6.000000E-02	7.000000E-02	1.945130E-07	1.000000E+00
7.000000E-02	1.200000E-01	4.937046E-07	1.000000E+00
1.200000E-01	1.700000E-01	4.394082E-07	1.000000E+00
1.700000E-01	2.200000E-01	3.709209E-07	1.000000E+00
2.200000E-01	2.700000E-01	3.133230E-07	1.000000E+00
2.700000E-01	3.200000E-01	2.648119E-07	1.000000E+00
3.200000E-01	3.700000E-01	2.243125E-07	1.000000E+00
3.700000E-01	4.200000E-01	1.925478E-07	1.000000E+00
4.200000E-01	4.700000E-01	1.655995E-07	1.000000E+00
4.700000E-01	5.200000E-01	1.457540E-07	1.000000E+00
5.200000E-01	5.700000E-01	1.284707E-07	1.000000E+00
5.700000E-01	6.200000E-01	1.113383E-07	1.000000E+00
6.200000E-01	6.700000E-01	1.006185E-07	1.000000E+00
6.700000E-01	7.200000E-01	8.809190E-08	1.000000E+00
7.200000E-01	7.700000E-01	7.899810E-08	1.000000E+00
7.700000E-01	8.200000E-01	7.067640E-08	1.000000E+00
8.200000E-01	8.700000E-01	6.385930E-08	1.000000E+00
8.700000E-01	9.200000E-01	5.758730E-08	1.000000E+00
9.200000E-01	9.700000E-01	5.363510E-08	1.000000E+00
9.700000E-01	1.020000E+00	4.816412E-08	1.000000E+00
1.020000E+00	1.070000E+00	4.336965E-08	1.000000E+00
1.070000E+00	1.120000E+00	3.917105E-08	1.000000E+00
1.120000E+00	1.170000E+00	3.593800E-08	1.000000E+00
1.170000E+00	1.220000E+00	3.283116E-08	1.000000E+00
1.220000E+00	1.270000E+00	2.958952E-08	1.000000E+00
1.270000E+00	1.320000E+00	2.742077E-08	1.000000E+00
1.320000E+00	1.370000E+00	2.473372E-08	1.000000E+00
1.370000E+00	1.420000E+00	2.298357E-08	1.000000E+00
1.420000E+00	1.470000E+00	2.098314E-08	1.000000E+00
1.470000E+00	1.520000E+00	1.946901E-08	1.000000E+00
1.520000E+00	1.570000E+00	1.712784E-08	1.000000E+00
1.570000E+00	1.620000E+00	1.573579E-08	1.000000E+00
1.620000E+00	1.670000E+00	1.440433E-08	1.000000E+00
1.670000E+00	1.720000E+00	1.274980E-08	1.000000E+00
1.720000E+00	1.770000E+00	1.161596E-08	1.000000E+00
1.770000E+00	1.820000E+00	9.791930E-09	1.000000E+00

```
  1.820000E+00  1.870000E+00  8.685100E-09  1.000000E+00
  1.870000E+00  1.920000E+00  7.154960E-09  1.000000E+00
  1.920000E+00  1.970000E+00  5.253931E-09  1.000000E+00
  1.970000E+00  2.000000E+00  2.067981E-09  1.000000E+00
-------------------------------------------------------------
```

Detector Files

The actual files are between the "----" lines.

DQE File: bgo_2cm_dqe_example2.dat

This file is 25 lines long.

```
   Energy                      DQE                      Energy Response

-------------------------------------------------------------------
  0                           1                        0
  2.000000000000000E-002  0.966831170880505    1.926269616719999E-002
  4.000000000000001E-002  0.991554414282508    3.957458097219003E-002
  6.000000000000001E-002  0.992573191546373    5.947049955983003E-002
  8.000000000000002E-002  0.992227641380919    7.926516008439997E-002
  9.000000000000002E-002  0.992652864706663    8.916088764671000E-002
  9.100000000000003E-002  0.855002441105997    7.560888856829993E-002
  0.100000000000000       0.891455975160858    8.591263660827998E-002
  0.300000000000000       0.975475735891477    0.290221932490300
  0.500000000000000       0.794738866850661    0.390825987501000
  0.700000000000000       0.643001019610579    0.434142030491000
  0.900000000000000       0.551165326871547    0.468777485865000
  0.910000000000000       0.548148547133081    0.471017842628000
   1.00000000000000       0.518535440430495    0.485402353210000
   3.00000000000000       0.370905706189826    0.963500151000000
   5.00000000000000       0.374417923903217    1.63490315200000
   7.00000000000000       0.38149460735669     2.33695588500000
   9.00000000000000       0.396511513837843    3.10747970000000
   11.0000000000000       0.409950910376004    3.88395039100000
   15.0000000000000       0.42954500524025     5.38514675000000
   20.0000000000000       0.45115976830741     7.20986689000000
   40.0000000000000       0.478487460450078    12.6218611200000
   60.0000000000000       0.48373910057121     16.0440545000000
   80.0000000000000       0.48490462799775     18.4117836300000
   100.000000000000       0.4844979602990      20.1884702800000
-------------------------------------------------------------------
```

Blur File

This file is 42 lines long.

```
-------------------------------------------------------------
3 50
 2.000000E-02 2.000000E-01 2.000000E+00
```

```
 4.000000E-03 4.663881E-03 5.437947E-03 6.340484E-03 7.392816E-03
 8.619803E-03 1.005043E-02 1.171851E-02 1.366343E-02 1.593116E-02
 1.857525E-02 2.165819E-02 2.525281E-02 2.944402E-02 3.433086E-02
 4.002876E-02 4.667234E-02 5.441856E-02 6.345042E-02 7.398131E-02
 8.626001E-02 1.005766E-01 1.172693E-01 1.367325E-01 1.594261E-01
 1.858861E-01 2.167376E-01 2.527096E-01 2.946519E-01 3.435554E-01
 4.005754E-01 4.670590E-01 5.445769E-01 6.349604E-01 7.403450E-01
 8.632202E-01 1.006489E+00 1.173536E+00 1.368309E+00 1.595407E+00
 1.860197E+00 2.168935E+00 2.528913E+00 2.948638E+00 3.438024E+00
 4.008633E+00 4.673947E+00 5.449684E+00 6.354169E+00 7.408772E+00
 1.903890E-02 3.548780E-05 2.564520E-05 1.736400E-05 1.084570E-05
 6.222280E-06 3.192930E-06 1.445640E-06 5.512390E-07 1.893180E-07
 5.088500E-08 1.204990E-08 1.259520E-09 2.627400E-10 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 1.229390E-01 1.409840E-03 9.948290E-04 7.289510E-04 6.556360E-04
 6.918800E-04 7.697490E-04 8.540320E-04 9.443760E-04 1.036160E-03
 1.145920E-03 1.253360E-03 1.360840E-03 1.464280E-03 1.561160E-03
 1.656150E-03 1.727030E-03 1.779480E-03 1.801990E-03 1.806750E-03
 1.772410E-03 1.700620E-03 1.581850E-03 1.432530E-03 1.258680E-03
 1.066590E-03 8.618680E-04 6.635740E-04 4.894640E-04 3.359730E-04
 2.193810E-04 1.339690E-04 7.439520E-05 3.546940E-05 1.549960E-05
 6.407030E-06 2.361010E-06 6.793690E-07 1.511930E-07 2.129270E-08
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 5.393080E-02 4.678000E-03 5.396440E-03 6.195130E-03 7.101940E-03
 8.149930E-03 9.276970E-03 1.051530E-02 1.183810E-02 1.324430E-02
 1.461590E-02 1.592020E-02 1.704310E-02 1.788610E-02 1.819990E-02
 1.789970E-02 1.666850E-02 1.455260E-02 1.156820E-02 8.338640E-03
 5.452930E-03 3.648080E-03 2.960730E-03 2.908350E-03 3.105690E-03
 3.331270E-03 3.569280E-03 3.781810E-03 4.019140E-03 4.200670E-03
 4.380390E-03 4.461790E-03 4.509260E-03 4.492190E-03 4.398220E-03
 4.197910E-03 4.002400E-03 3.676940E-03 3.268470E-03 2.781640E-03
 2.327780E-03 1.881810E-03 1.420650E-03 1.031920E-03 6.955310E-04
 4.465830E-04 2.552740E-04 1.394130E-04 6.877590E-05 2.980920E-05
-------------------------------------------------------------------
```

3. Neutron radiograph of a meshed object

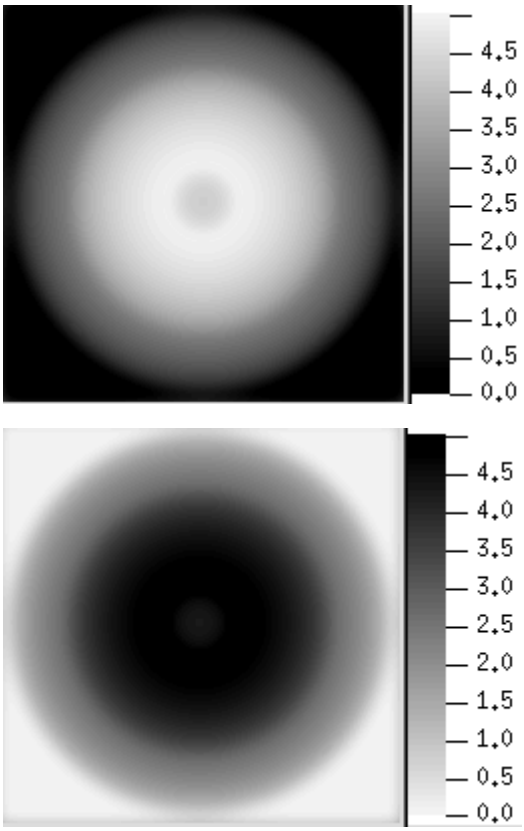
In this example, we radiograph two nested shells, of tungsten and copper, with a 3 MeV neutron source of finite size. In this case, the nested shells are specified in a 1DrASCII object. This is a 1D radial mesh, stored in an ASCII formatted file. The format of this mesh is displayed below. This neutron source was chosen to mock up a D+D neutron generator, but with a broad Gaussian spatial distribution. In order to have readily observeable blur, we have given the source an 8 mm FWHM. Note that the path lengths through the objects are smaller

than in the X-Ray cases. Note also that the outer shell of Copper is more observable with neutrons.
Obtained by typing *hades example3.hades example3.mesh*

Input Deck and Radiograph

```
//Start of deck example3.hades
verbose on
preview on
neutron energy = 3
FWHM spot 0.8
ifmax = 200
jfmax = 200
x1    = +20
y1    = 200
z1    = -20
x2    = -20
y2    = 200
z2    = -20
x3    = -20

z3    = +20
xs    = 0.
ys    = -100
zs    = 0.
tiff on
outtiff example3
outview example3
object type 1DrASCII
//End of deck
```



1DrASCII Mesh File example3.mesh

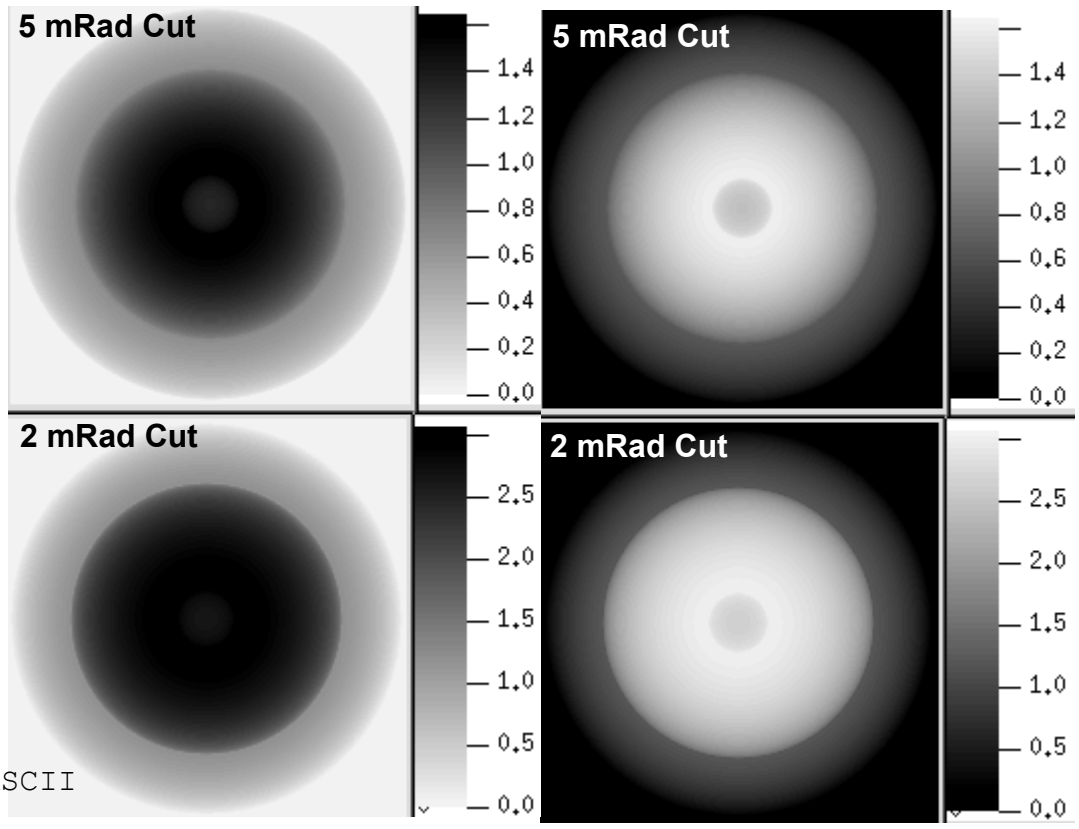
This file is 6 lines long.

```
3 2                          Number of Zones, Number of Materials
  0.00    1.0   1
 19.3     4.5   1           Zone Info:  Density, Outer Radius, Material Number
  8.96    6.5   2
74000  1.0                   Material Info:  Z*1000+A, Number Fraction
29000 1.0
```

4. Perfect Lensing Proton Radiography for a 1DrASCII mesh

In this example, we radiograph two nested shells, of tungsten and copper, with 23.01 GeV protons. Perfect lensing is assumed for the imaging. The object is again specified with the same 1D radial mesh as in Example 3. Note that the path lengths through the objects are smaller than in both the x-ray and neutron cases. Reducing the anguler cut causes the pathlength to increase, because more scattered protons are being excluded from the image.
Obtained by typing *hades example4.hades example3.mesh*

```
//Start of deck example4.hades
verbose on
preview on
protons
  kenergy 23.01
  cut 5
  cut 2
endprotons
ifmax = 200
jfmax = 200
x1    = +20
y1    = 200
z1    = -20
x2    = -20
y2    = 200
z2    = -20
x3    = -20
y3    = 200
z3    = +20
xs    = 0.
ys    = -100
zs    = 0.
tiff on
outtiff example4
outview example4
object type 1DrASCII
//End of deck
```



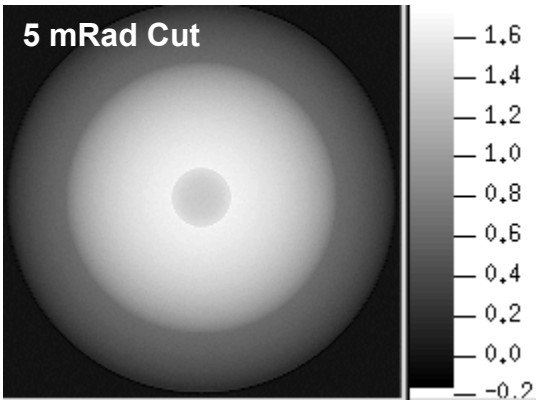
5. Proton Radiography Using the Brookaven National Laboratory U Line

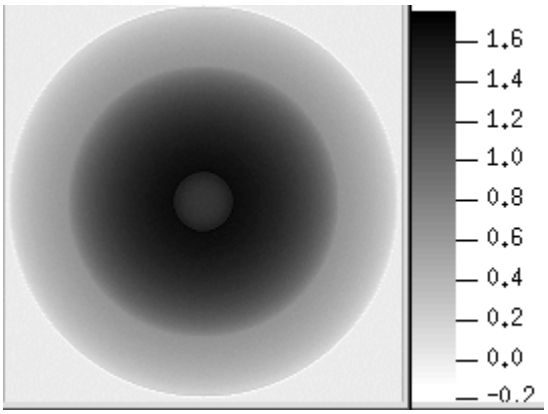
In this example, we simulate proton radiography as in example 4, except that perfect lensing is not assumed. In this case transport through the U Line at BNL is modeled. The same angular cuts are used, but the more complex transport of the protons results in slightly different proton attenuation and also a sharpening of the image near interfaces. This effect is called limbing. There is also a slight warping of the image due to the action of the lenses. Obtained by typing *hades example5.hades*

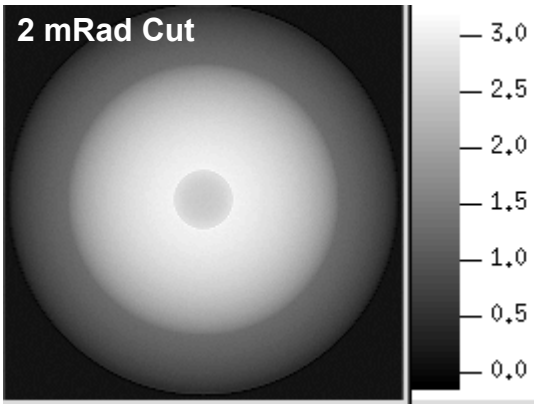
Input Deck and Radiograph

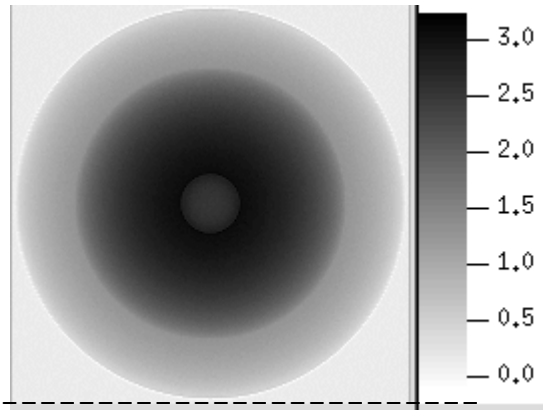
```
//Start of deck example5.hades
protons
  prad beamline data file E955_prad_geometry.dat
  cut 5
  cut 2
endprotons
verbose on
preview on
ifmax = 200
jfmax = 200
```

```
x1     = +6.666667
y1     = 200
z1     = -6.666667
x2     = -6.666667
y2     = 200
z2     = -6.666667
x3     = -6.666667
y3     = 200
z3     = +6.666667
xs     = 0.
ys     = -100
zs     = 0.
excluded object  //Shell of Copper
  xcenter = 0; ycenter = 0; zcenter = 0
  sphere // Shell of Copper
    x = 0.0; y = 0; z = 0.0
    radius = 6.5
  endsphere
  sphere
    x = 0.0; y = 0; z = 0.0
    radius = 4.5
  endsphere
  difference
  isofrac = 29000 1.0
  rho = 8.96
endexcluded object
excluded object  //Shell of Tungsten
  xcenter = 0; ycenter = 0; zcenter = 0
  sphere
    x = 0.0; y = 0; z = 0.0
    radius = 4.5
  endsphere
  sphere
    x = 0.0; y = 0; z = 0.0
    radius = 1
  endsphere
  difference
  isofrac = 74000 1.0
  rho = 19.3
endexcluded object
// End of deck
```









Beamline Specification Deck

This file is 43 lines long and has a blank line at the end.

```
23.09                          ! Beam energy in GeV
 5                 ! FWHM in cm
1.75009269E-03 -1.75009429E-03 ! mottershead u and v
27                 ! number of gaps and magnets
 9                 ! number of gaps and magnets upstream of the object location
```

```
 1                    ! number of image planes
27                    ! number of gaps and magnets upstream of each image plane
60.                   ! length in cm of the collimators
1                     ! number of collimators
18                    ! number of gaps and magnets upstream of each collimator
!
! the list of magnets and gaps
!
Item      Aperture   Field Strength   Length
  #        (cm)        (kGauss)       (cm)
 Gap1       0           0           1457.662
  Q0       3.81        7.549964      94.472
 Gap2       0           0           138.105
  Q1       3.81        7.549964      94.472
 Gap3       0           0           412.66
  Q2      10.16       -7.793271      130.8
 Gap4       0           0            30
  Q3      10.16       -7.793271      130.8
 Gap5       0           0           250
 GapO       0           0           200
  Q4      10.16       11.8245158     130.8
 Gap6       0           0            30
  Q5      10.16       11.8245158     130.8
 Gap7       0           0            65
  Q6      10.16      -11.8245158     130.8
 Gap8       0           0            30
  Q7      10.16      -11.8245158     130.8
 Gap9       0           0           200
 GapC       0           0           200
  Q8      10.16       11.8245158     130.8
 Gap10      0           0            30
  Q9      10.16       11.8245158    130.8
 Gap11      0           0            65
 Q10      10.16      -11.8245158     130.8
 Gap12      0           0            30
 Q11      10.16      -11.8245158     130.8
 Gap13      0           0           200
```

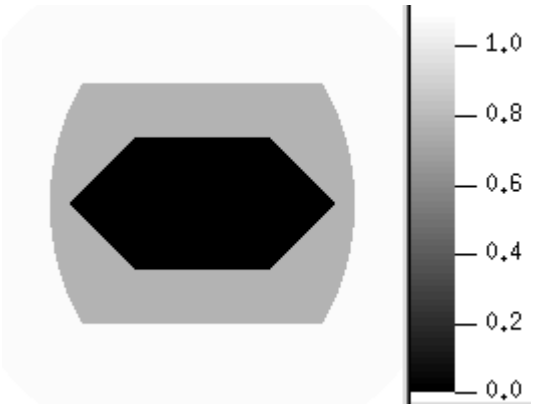
---

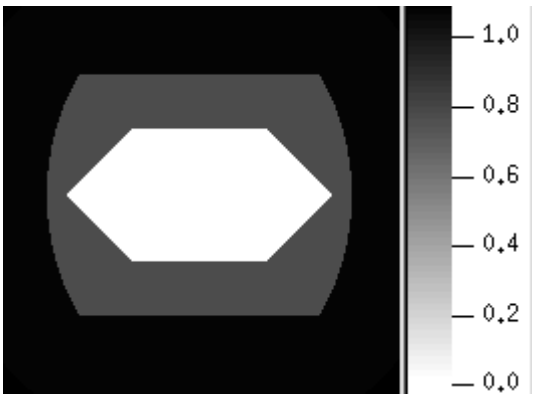
6. Example of Generalized Collimators and a Mixture by Weight %

Specifying generalized collimators is a delicate art, so we have included an example that has two of these collimators specified. There is no object specified, but the magnification is the same as Examples 1-4. The first collimator starts halfway between the source and (0,0,0) and is 10 cm thick. For variety, we have made this object out of paraffin wax and specified the compositions by weight %, denoted by negative signs in front of the mass fractions. This collimator is specified by two horizontal cuts intersecting two vertical circular arcs. The second object is a hexagonal hole cut out of a 1 cm thick plate of tungsten and starts 90 cm from the source. Note that the horizontal cuts in both collimators have the same width. Their positions determine how broad the actual shadows are. Each cut is sharp because we have specified the back curves so that they exactly match the cone beam of the source.
Obtained by typing *hades example6.hades*

```
//Start of deck example5.hades
verbose on
```

```
preview on
gamma energy = 4
ifmax = 200
jfmax = 200
x1    = +10
y1    = 200
z1    = -10
x2    = -10
y2    = 200
z2    = -10
x3    = -10
y3    = 200
z3    = +10
xs    = 0.
ys    = -100
zs    = 0.
tiff on
outtiff example10
gencoll
  yfront = -50; yback = -40
  segmentf 1 1 -1 1
  arcf 0.73205 0 2
  segmentf -1 -1 1 -1
  arcf -0.73205 0 2
  segmentb 1.2 1.2 -1.2 1.2
  arcb  0.87846 0 2.4
  segmentb -1.2 -1.2 1.2 -1.2
  arcb -0.87846 0 2.4
  rho 0.93 // Specify Paraffin wax using weight %
  isofrac 1001 -0.148605 6000 -0.851395
endgencoll
gencoll
  yfront = -10; yback = -9
  segmentf 1 1 -1 1
  segmentf -1 1 -2 0
  segmentf -2 0 -1 -1
  segmentf -1 -1 1 -1
  segmentf 1 -1 2 0
  segmentf 2 0 1 1
  segmentb 1.011111 1.011111 -1.011111 1.011111
  segmentb -1.011111 1.011111 -2.022222 0
  segmentb -2.022222 0 -1.011111 -1.011111
  segmentb -1.011111 -1.011111 1.011111 -1.011111
  segmentb 1.011111 -1.011111 2.022222 0
  segmentb 2.022222 0 1.011111 1.011111
  rho 19.3
  isofrac 74000 1.0
endgencoll
// End of deck
```





Appendix B: Reading .sdt Files with IDL

The following IDL procedure will allow the user to read in a View-formatted file and will return a 2D array for manipulation in IDL. The procedure would be invoked in IDL as follows:
vread,'image',array,header,
where *array* is the variable which will contain the image, *header* is a data structure which will contain the header information read in from the .spr file, and *image* is the name of the View formatted file which is to be read in. Do not include an ".sdt" or ".spr" suffix.

```
pro vread,file,img,hdr, swap = swap

    if(n_params() ne 3) then begin
      message, "Calling sequence: vread(file,img,hdr)"
      goto, EndOnErr
    endif

    hfile=file+'.spr'
;    print
;    print,' vread opening header file: ', hfile
     openr,69,hfile

    hdr=fltarr(8)
    readf,69,hdr
    close,69

;    print,'--------'
;    print,'      number of dimensions: ',hdr[0]
;    print,'        number of x-pixels: ',hdr[1]
;    print,' x-position of pixel (0,0): ',hdr[2]
;    print,'          x-width of pixel: ',hdr[3]
;    print,'        number of y-pixels: ',hdr[4]
;    print,' y-position of pixel (0,0): ',hdr[5]
;    print,'          y-width of pixel: ',hdr[6]
;    print,'          .sdt file format: ',hdr[7]
;    print,'--------'

    dfile=file+'.sdt'
;    print,' vread opening data file: ', dfile
    if hdr[7] EQ 1 then begin
      print,'   File is composed of 16 bit words (signed integers)'
      img = intarr(hdr[1],hdr[4])
    endif
    if hdr[7] EQ 3 then begin
      print,'   File is composed of 32 bit reals'
      img = fltarr(hdr[1],hdr[4])
    endif
    if hdr[7] NE 1 AND hdr[7] NE 3 then begin
      message, " Cannot handle this data type"
```

```
      goto, EndOnErr
    endif

    openr,69,dfile
    readu,69,img
    if(keyword_set(swap)) then img = swap_endian(img)
    close,69

;    print,img[0:5]

    EndOnErr: Begin
    End
end
```

Appendix C: Detector Definition

By default, HADES images are two-dimensional arrays of -ln(I/I_o), where both the attenuated and unattenuated intensities are in units of dose. In practice, the energy response of a given detector is not necessarily proportional to the dose fluence, rather, the energy response may be flat (as for a thin detector), proportional to photon energy, as for a thick scintillator, or complicated, as for a detector material with photon absorption edges in the energy range of interest. Furthermore, both the image blurring and the detector quantum efficiency (DQE) vary with photon energy. (The DQE, for our purposes here, is approximately the probability that the photon will deposit energy in the detector. This probability, always less than unity, increases the calculated noise level above what one would expect given √N statistics.) Another important quantity for modeling detector response is what we call here the "energy response". This quantity is how much of a given photon's energy is actually deposited in the active detector region, generating a signal. The reasoning behind this treatment of detectors is documented elsewhere[3].

There are two detector files that the user may supply to HADES to model the response of a given, non-ideal detector. The first file, the simpler of the two, gives both the energy-dependent DQE and the energy dependent response of the detector. The form of the command is:

`dqe data file` [filename]

This file contains three columns of numbers, with each row containing:

Energy (MeV) DQE(at this energy) Response(At this energy)

The DQE and response curves are linearly interpolated within HADES as need be. The actual entries in each line are free-format. The energy is assumed to be monotonically increasing. There is no requirement that the energy mesh given the detector response file be (a) evenly spaced, (b) be the same as the energy mesh in the photon spectrum used for the particular radiograph, or (c) be the same as used in the detector blur file, described below. This freedom allows the user to use a finely-spaced energy mesh around photon absorption edges, where the DQE and the energy response can vary dramatically. This freedom also allows the user to generate the detector data without having to know what energy spectra may be used later. A sample data file, (a severely thinned version of a BGO detector file) would be:

```
0       1           0
0.02    0.9668      0.0192
0.04    0.9915      0.0395
0.06    0.9925      0.0594
0.08    0.9922      0.0792
0.09    0.9927      0.0892
0.091   0.8550      0.0756
0.1     0.8914      0.0859
0.3     0.9754      0.29
0.5     0.7947      0.39
```

[3] A. E. Schach von Wittenau, C. M. Logan, M. B. Aufderheide and D. M. Slone, Med. Phys. **29 (11)**, Nov. 2002, pp. 2559-2570.

```
  0.7    0.6430      0.43
  0.9    0.5511      0.46
  1      0.5185      0.48
  3      0.3709      0.96
  5      0.3744      1.63
  7      0.3814      2.33
  9      0.3965      3.10
 11      0.4099      3.88
 15      0.4295      5.38
 20      0.4511      7.20
 40      0.4784     12.62
 60      0.4837     16.04
 80      0.4849     18.41
100      0.4844     20.18
```

Note the rapid change in DQE and energy deposition at 0.09MeV (the K-edge for the Bismuth in the BGO).

The energy-dependent detector blur is given by an additional data file, specified to HADES via:

`detector blur file` [filename]

The form of this file is

Line 1 => [Number Of Energies] [Number Of Radii]

Freeform block of energy values

Freeform block of R values

Freeform block of blur(R,E) data (R index varying first)

The blur(R,E) values are obtained, for example, via Monte Carlo simulation. For a thick scintillator, the Monte Carlo user would specify a monolithic slab of the scintillator material, and segment the energy deposition tally using nested cylinders. The Monte Carlo code then returns a list of energy deposits in each cylinder, perhaps in the form

Cylinder #	Outer Radius	Energy Deposit
1	R1	E1
2	R2	E2
	…	…
	NumberOfRadii	E_{NoR}

The Monte Carlo user would calculate such data at photon energies of interest, and transfer the radial values for the bounding cylinders (which must be the same for all energies) and the deposited energy values into the blur file. For film packs, the Monte Carlo user would calculate the energy deposited in the emulsion layer.

A sample blur file is given below. The first line indicates that there 78 sets of blur calculations, each of which contains 100 points. The next several lines give the 78 energy values, in this case ranging from 20 keV to 100 MeV. The next set of lines gives the radii of the tally cylinders used in the underlying MCNP simulations; here, the tally cylinder radii range from 0.004 cm to 8 cm. Finally, the results of the energy deposition tallies (in MCNP, a *F8 tally) are given for each energy. The lines below were taken from a file describing a 1-mm thick LSO detector.

```
78 100
 2.000000E-02 3.000000E-02 3.500000E-02 4.000000E-02 4.500000E-02
 5.000000E-02 5.500000E-02 6.000000E-02 6.300000E-02 6.400000E-02
   several lines of energy values removed
 1.900000E+01 2.000000E+01 2.500000E+01 3.000000E+01 3.500000E+01
 4.000000E+01 4.500000E+01 5.000000E+01 5.500000E+01 6.000000E+01
 6.500000E+01 7.000000E+01 7.500000E+01 8.000000E+01 8.500000E+01
 9.000000E+01 9.500000E+01 1.000000E+02
 4.000000E-03 4.319204E-03 4.663881E-03 5.036064E-03 5.437947E-03
 5.871900E-03 6.340484E-03 6.846461E-03 7.392816E-03 7.982770E-03
   several lines of radius values removed
 2.730723E+00 2.948638E+00 3.183942E+00 3.438024E+00 3.712382E+00
 4.008633E+00 4.328527E+00 4.673947E+00 5.046933E+00 5.449684E+00
 5.884574E+00 6.354169E+00 6.861238E+00 7.408772E+00 8.000000E+00
 1.960440E-02 1.758330E-05 1.522930E-05 1.328480E-05 1.127790E-05
 9.620540E-06 8.058520E-06 6.704050E-06 5.515760E-06 4.538090E-06
 3.656850E-06 2.898950E-06 2.257210E-06 1.760840E-06 1.300950E-06
 1.017960E-06 7.374470E-07 5.237870E-07 3.690990E-07 2.580710E-07
 1.555870E-07 1.056750E-07 6.095890E-08 3.762950E-08 2.038790E-08
 1.131340E-08 5.935870E-09 1.754830E-09 1.771360E-09 1.180900E-09
 3.936350E-10 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
    the rest of the lines for this energy have been removed.
    The *F8 results from the 30 keV simulation are given next.
 2.901190E-02 4.899720E-05 4.722460E-05 4.569290E-05 4.399300E-05
 4.223200E-05 4.080810E-05 3.904120E-05 3.726530E-05 3.547000E-05
 3.384500E-05 3.216240E-05 3.005530E-05 2.819340E-05 2.628940E-05
 2.430400E-05 2.226320E-05 2.028830E-05 1.828910E-05 1.611400E-05
 1.428190E-05 1.229500E-05 1.059730E-05 9.015180E-06 7.490450E-06
 6.088170E-06 4.972640E-06 3.922630E-06 3.032570E-06 2.293840E-06
 1.685580E-06 1.220800E-06 8.487750E-07 5.760610E-07 3.527580E-07
 2.366010E-07 1.352050E-07 8.385490E-08 4.559820E-08 2.669520E-08
 1.780110E-08 7.550180E-09 2.424800E-09 2.020670E-09 8.082680E-10
 4.041340E-10 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
     more lines have been removed
```

Appendix D: Post Processing

HADES produces a variety of output files. The code will always generate a text file (see `logfile`) that contains the same runtime information that is output to the screen, in addition to other, more detailed information about the simulation.

When `preview` is set, a Portable GrayMap file is generated (see `outview`) and automatically displayed. The resulting `.pgm` file can be redisplayed using **`xv`**, available on many Unix machines, by typing: **`xv`** `xray.pgm`.

When `tiff` is set, a Tagged Image File Format file is generated (see `outtif`). The resulting `.tif` file can be displayed using **`xv`**, available on many Unix machines, by typing: **`xv`** `xray.tif`. TIFF files are often used to exchange images between B Division and groups elsewhere.

It should be emphasized that HADES computes and returns images whose units are $-\ln(I/I_o)$, which are units of pathlength or effective pathlength, and are floating point numbers rather than integers. PGM and TIFF image formats require 8 bit integers. When such a format is requested, HADES finds the min and max values of the image and scales the image so that the min corresponds to a value of 0 and the max corresponds to a value of 255. Information is lost in this conversion (both in terms of absolute scale and precision), but for qualitative work, little has been lost.

When `pov` is set, a POV-Ray file is generated (see `outvis`). The resulting `.pov` file can be rendered using **`povray`**, by typing: **`povray`** `xray.pov`. Since a HADES radiograph is a 2D projection of a 3D scene, one can use the POV-Ray renderer to better understand any overlapping or occluding effects that are represented in the radiograph. Two visualization files are generated. The first is taken along the source-detector axis and includes all the objects, the detector, and a beam representing the probe. This file is called `xray.pov` and is left-right reversed relative to the radiograph, i.e. the z-axis is flipped. This is because HADES is looking from behind the detector toward the source, the opposite of the pov scene. The second file is called xray_o.pov and is oriented typically along the source-detector plane and perpendicular to the objects. This scene shows the objects in depth and also shows the source and detector location in space.

HADES will generate a silo file by default (see `outsilo`). The resulting `.silo` file can be displayed using **`MeshTV`** or VisIt (visualization programs used at LLNL), by typing: **`meshtv,`** Opening `xray.silo`, and choosing `Plots->Psuedocolor->Image`, and `Draw`. Note that the `Image` plot is of the final, processed radiograph. All intermediate radiographs are available under the same `Plots->Psuedocolor` menu.

We have recently developed a Python GUI called CHARON, which is able to open .sdt files and do fairly standard simple image processing tasks (such as viewing, color map selection, conversion to TIFF, GIF or PGM, getting slices). This is becoming a popular way to view HADES images. Setting the `py` option creates an executable script that invokes CHARON on the SDT output file. CHARON can also be invoked explicitly by typing **`charon file1.sdt file2.sdt ….`**

Appendix E: Other HADES Invocations

Restarts

HADES may be restarted from any point in a previous run. This capability is especially useful on machines where interactive time limits may be not long enough to complete a radiograph from a problem containing a mesh with many domains. In order to allow for restarting, be sure to set the `dump` option (see `outres` for explicit naming of the restart file). The restart file is a SILO formatted file and is fully self-contained. This means that no input deck is necessary when restarting, nor need the mesh file be respecified. The command for restarting HADES is essentially the same as the initial command:

hades xray.res

A problem may be restarted as many times as needed, but remember that HADES automatically adds a .res extension to the new restart file each time, i.e. restarting with the file `xray.res` will generate a new restart file `xray.res.res`.

Parallel HADES

HADES may be run in parallel for efficiency. The finite spot source is parallelized, as are some computations involving meshes. In general, HADES does not do its own decomposition on a mesh; the code uses the decomposition defined in the mesh file.

HADES currently uses MPI to handle the parallel communication. Currently the code is not multi-threaded. HADES may be therefore be run in parallel only on the Compaq alphas and IBM SP2s. At LLNL, the Compaqs and IBMS support both interactive and batch processing.

To run interactively on the DECs type

`mpirun –np #procs /usr/local/apps/bin/hades deckname [meshname]`, or

`hades inputdeck [meshname] –np #procs`, or

`hades –np #procs inputdeck [meshname]`

where `#procs` is the number of cpus, `inputdeck` is the input deck, and `meshname` is the optional mesh file. `#procs` is limited to the number of available cpus on a specific machine, usually 8 or 12.

In order to run in batch mode, a batch script needs to be created and submitted to the batch system. See the `PSUB man` pages for more details. A minimal script for the DECs would be similar to:

```
#!/usr/ bin/csh
# set accounting bank
#PSUB –b bdivp
# set time limit
#PSUB –tM 300m
# set stdout and stderr to go to the batch output file
#PSUB –eo
# end of PSUB commands
```

```
#PSUB
# change to submit directory
chdir /home/user/somedir
# run hades with 8 processors, the output is captured in the batch output file
#mpirun –np 8 hades inputdeck [meshname]
hades inputdeck [meshname] –np 8
```

To submit this batch script, type

```
psub batchscript
```

To run interactively on the IBMs, type

```
poe –nodes #nodes –procs #procs –rmpool 0 /usr/local/apps/bin/hades
deckname [meshname]
```

On the IBMs, the batch script is essentially the same, but both the number of nodes and the number of processors (at four cpus per node) must be specified. A minimal batch script would be like:

```
#!/usr/ bin/csh
# set accounting bank
#PSUB –b bdivp
# set time limit
#PSUB –tM 300m
# set stdout and stderr to go to the batch output file
#PSUB –eo
# set number of nodes
#PSUB –ln 8
# set number of processors
#PSUB –g 32
# end of PSUB commands
#PSUB
# change to submit directory
cd /home/user/somedir
# run hades, the output is captured in the batch output file
hades inputdeck [meshname]
```

To submit this batch script, type

```
psub batchscript
```

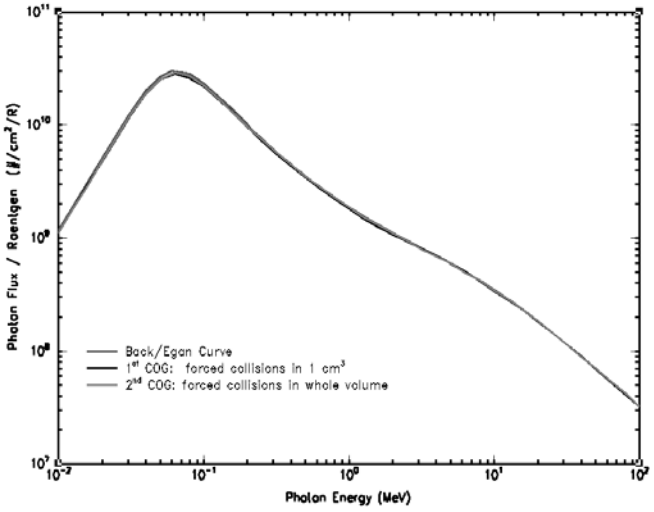
Appendix F: Photon Dose Conversion Curve

The strength of X-ray and Gamma ray sources is often given in terms of dose. While a useful concept for those initiated into the mysteries of photon physics, dose is one of the stumbling blocks for a novice in this field. HADES is able to convert from dose to flux for its calculations, freeing the user from sweating these details. In this section, we define dose and explain the dose conversion curve used by HADES. This discussion is drawn from Halmshaw's book on industrial radiography[4].

Dose was originally measured in units of Roentgens, which is the amount of radiation that produces in 1 cm^3 of dry air at 0^o C and 760 mm Hg pressure (that is, 0.001293 g of air) a total ionization corresponding to 1 esu of charge. This is an exposure because it counts the amount of ionization caused in the air. Metric units for exposure are Coulombs/kg, one of which is 3876 Roentgens.

One can also think in terms of the amount of energy deposition required to cause this amount of ionization in 1 cm^3 of air, which is the absorbed dose. 1 Roentgen corresponds to 83.3 ergs/g of energy deposition. In 1954, the ICRU defined a new unit of absorbed dose, the Rad, which corresponds to 100 ergs/g of absorbed dose. HADES works in units of Roentgens, but for most applications, the difference between Roentgens and Rads is not large enough to worry about.

HADES actually expects Dose specification in terms of Roentgens at 1 meter, since doses are often measured that distance from the source. HADES needs to convert dose in Roentgens into a flux, so that it knows how many photons are in each energy bin in each pixel. It does this using a dose conversion curve that is obtained by calculation. We calculate, for a uniform flux of photons one meter from a cubic cm of air, how much of the energy is absorbed in the volume of air of interest. For low energy photons, this calculation could be done essentially analytically, because the dominant process is photoelectric absorption, which captures the photon locally. For higher energies, Compton scattering and pair production can occur, resulting in partial energy deposition in the volume of interest. In addition, some energy is given to high-energy electrons, which may escape the measurement of volume. Because of these complications we resort to Monte Carlo calculations. We have used the COG Monte Carlo code[5] to do these calculations. At right is the curve used in HADES for dose conversion. None of these calculations used electron transport



[4] R. Halmshaw, *Industrial Radiography: Theory and Practice 2nd Edition*, London: Chapman and Hall, 1995, pp.238 and following.

[5] T. Wilcox and E. Lent, "COG – A Particle Transport Code Designed to Solve the Boltzmann Equation for Deep Penetration (Shielding) Problems, Vol. I Users Manual", LLNL Rept. M-221-1 (1989).

because it was assumed that electronic equilibrium exists between the measurement volume and the surrounding volume. Future work will be needed to extend the curve to higher energies, if necessary.

When the user specifies a dose and a single photon energy, this curve is used to convert to the flux of photons for this single photon energy. When the user specifies a number spectrum and a dose, HADES shifts the absolute normalization of this spectrum so that the total dose due to the spectrum equals the amount specified by the user. This is done by converting between dose and flux for each energy bin.

It should be emphasized that there is quite a difference, statistically, between a monochromatic shot with a specified dose and a spectral shot specified with the same dose. In the monochromatic case, every photon in the problem will be used in generating an image. In the spectral case, the total dose is spread over the spectrum. Most of the low energy photons will not penetrate the object, being effectively wasted. Thus the spectral image will look noisier than the monochromatic shot, because less photons will be available at the optimal energies.

Appendix G: Data Used and Locations

Location of data files

HADES selects one or two of four data files, depending on the type of calculation requested. For X-ray/Gamma Ray simulations, the *epdlbin* file is used. These contain the total photon absorption cross sections from LLNL's Evaluated Photon Data Library[6].

For neutron simulations, the *neutron_sigmas* file is used.. These are total absorption cross section for neutr from LLNL's Evaluated Neutron Data File[7].

For proton radiography simulations, two data files are used: *ppelastic.dat*, and *pptotal.dat*. These files reside in `/usr/local/apps/data/hades` on our local machines[8]. The Letaw et al. fits for proton-nuclear cross sections is given below[9]. Dahl's fit for radiation length is a staple of particle physics and is given in all Particle Data Group publications[10]. For proton energy loss, we are currently switching from a simple Bethe-Bloch formula to the tables of Ziegler et al.[11] for 1 eV to 6 GeV, with an augmented Bethe-Bloch for higher energies. By the time this manual is publicly available, the transition will be complete.

The user may specify the use of a local data file in two ways. First, HADES checks the local directory to see whether the needed data file is there. If so, HADES uses the local data file, rather than the common file. Second, the `HADESPATH` environment variable can be used to set the correct directory. Note that not all the data files must be local, it is perfectly acceptable to keep a local version of *pptotal.dat* for example, and use the public versions of the other data files.

---

[6] D. E. Cullen et al., "Tables and Graphs of Photon-Interaction Cross Sections from 10 eV to 100 GeV Derived from the LLNL Evaluated Photon Data Library (EPDL), Parts A and B, UCRL-50400, Vol. 6A and 6B, Rev 4, 1989. David Resler generated the tables used in HADES in ASCII format from EPDL in 1996.

[7] R. J. Howerton et al., "The LLL Evaluated Nuclear Data Library (ENDL): Graphs of Cross Sections from the Library", UCRL050400, Vol. 15, Part B. 1976. David Resler generated the tables used by HADES in ASCII format from ENDL in 1997.

[8] The PP total and elastic cross sections were obtained from the Particle Data Group's website in 1998. The current address for these cross sections is http://pdg.lbl.gov/2002/contents_plots.html, but the cross sections at this site are more up to date that the default HADES files. There is not likely to have been much of a change.

[9] J. R. Letaw, R. Silberberg, and C. H. Tsao, *Ap. J. Supp.*, **51**, (1983) p.271.

[10] Particle Data Group, "Review of Particle Properties", *Physical Review*, **D66**, (2002), p.199.

[11] J. F. Ziegler, J. P, Biersack, U. Littmark, "The Stopping and Range of Ions in Solids," vol. 1 of series *Stopping and Ranges of Ions in Matter*, Pergamon Press, New York (1984). See also http://www.srim.org for Ziegler's most recent work.

Index of HADES Keywords

2

2dALExy_zbottom 20
2dALExy_ztop 20

A

A (ellipsoid) 30
angles 37
arcb (gencoll) 31
arcf (gencoll) 31
autotune 26

B

B (ellipsoid) 30
back file 18
background_energy_spectrum_file 24
background_level 23
background_radial_level_file 24
batch (DEC) 57
batch (IBM) 58
block commands (definition) 24

C

C (ellipsoid) 30
camera 37
catt 39
cback 39
ccg 18
cfront 39
collimator 30
comments 14
comments (in mesh material files) 38
cone 30
crho 39
cut (protons) 25
cut (used in an example) 47

D

DEDX 28
delta 37
detector blur file (command) 22
detector blur file (format of file) 54
detector blur file (used in an example) 41
diagnostics 19
difference 34
dn 21
Dose 23
Dose (used in an example) 42
dqe data file (command) 22
dqe data file (format of file) 53
dqe data file (used in an example) 41
dump 18, 57

E

ellipsoid 30
endcamera 38
endcollimator 30
endcone 30
endellipsoid 30
endexcluded object 33
endgencoll 31
endmulti_thetaphi 36
endmultiangle 35
endmultiphi 35
endmultitheta 36
endpairs (spot) 27
endpairs (zrprofile) 33
endplate 29
endprotons 25
endsinogram 37
endsphere 29
endspot 27
endtorus 33
endzrprofile 32
eps (torus) 33
eps (zrprofile) 33
excluded object 33
excluded object (used in an example) 42

F

flip 20
flip (gencoll) 32
FOV 38
FWHM (protons) 26
FWHM film 22
FWHM spot 22
FWHM spot (used in an example) 46

G

gamma energy 21
gamma energy (used in an example) 40
gencoll 31
gencoll (used in an example) 50
GLOBAL ANGLE TYPE (solid body object) 28
GLOBAL PHI 28
GLOBAL THETA 28
GLOBAL X 28
GLOBAL Y 28
GLOBAL Z 28

H

HADESPATH 61
height (cone) 30

I

idflwt 21
ienb 21
ifmax 16, 37
Image padding 22
index 37
intensity (protons) 26
intersection 34
invsqr 22
isofrac (general syntax) 38
isofrac (syntax for mesh materials) 38
isofrac (syntax for solid-body objects) 38
isofrac (use with meshes) 20
isofrac (use with solid body objects) 28
isofrac (used in an example) 40
isofrac (used with mass fractions) 50

J

jfmax 16, 37

K

keepinside 21
keepoutside 21
kenergy (protons) 25
kenergy (used in an example) 47

L

label 28
lateral blur (protons) 25
li (collimator) 31
lo (collimator) 31
LOCAL ANGLE TYPE (mesh) 17, 37
LOCAL ANGLE TYPE (solid body object) 27
log term (protons) 25
logfile 17
logfile (reference in Post Processing Appendix) 56

M

mac 20
mac (solid body object) 28
MAG 38
matcol 39
material multiply 20
maxhist 19
maxtif 18
mesh center X 17
mesh center Y 17
mesh center Z 17
minhist 18
mintif 18
mpirun 57
multi_thetaphi 36
multiangle 35
Multiangle Mode 35, 36, 37
multiphi 35
multiprojection 28
multitheta 36
MUp 28

N

naminp 20
nbeamlets 26
nbin 18
neutron energy 11, 21
neutron energy (used in an example) 46
Neutron Radiography 21
noobject 19
nplot 19
Nuclear Only (protons) 25
num_phi 37
num_theta 36
number of pairs (torus) 33
Number of Phi Zones 27
number spectrum 21
number spectrum (used in an example) 41

O

object type 19
object type (used in an example) 46
offset (collimator) 31
orientation 17, 37
ORIENTATION 14
outhist 18
outpy 18
outres 18, 57
outsilo 17
outsilo (reference in Post Processing Appendix) 56
outtif 18
outtif (reference in Post Processing Appendix) 56
outview 17
outview (reference in Post Processing Appendix) 56
outvis 18
outvis (reference in Post Processing Appendix) 56

P

pairs (spot) 27
pairs (zrprofile) 33
Parallel Beam 21
parallel execution 57
pdata (protons) 25
per energy 22
phi 17
phi (multi_thetaphi) 36
phi (multiangle) 35
phi (multiphi) 35
phi (multitheta) 36
phi (multitheta) 36
phi (sinogram) 37
phi (solid body object) 28
phi_end (multiphi) 36
phi_end (multiphi_phi) 37
phi_start (multiphi) 35
phi_start (multiphi_phi) 36
plate 29
poe 58
pov 18
pov (reference in Post Processing Appendix) 56
prad beamline data file (used in an example) 47
preview 19
preview (reference in Post Processing Appendix) 56
proton radiography beamline 26
protons 25
protons, full beamline (used in an example) 47
protons, perfect lensing (used in an example) 47
py 18, 56

R

radius (cone) 30
radius (sphere) 29
Radius (torus) 33
RadLen Only (protons) 25
rback 39
refx 20
refy 20
refz 20
restarting execution 57
revtif 18
rfront 39
rho 28
ri (collimator) 31
ro (collimator) 31

rot_num (multiphi) 36
rot_num (multitheta) 36
row 37

S

seed (protons) 26
segmentb (gencoll) 32
segmentf (gencoll) 31
siloimage 17
sinogram 37
skipcyc 19
solid body blocks 27
spectrum at 23
spectrumfile 21
spectrumfile (used in an example) 41
sphere 29
sphere (used in an example) 40
spot 27
SPOT type 22
start 37
stop 37
swap 17

T

theta 17
theta (multi_thetaphi) 36
theta (multiangle) 35
theta (multiphi) 35
theta (multiphi) 35
theta (multitheta) 36
theta (sinogram) 37
theta (solid body object) 28
theta_end (multi_thetaphi) 36
theta_end (multitheta) 36
theta_start (multi_thetaphi) 36
theta_start (multitheta) 36
tiff 18
tiff (reference in Post Processing Appendix) 56
torus 33

U

union 34

V

verbose 19
viewimage 17

X

x (collimator) 30
x (cone) 30
x (ellipsoid) 30
x (sphere) 29
x (torus) 33
x (zrprofile) 33
X0 28
x1 16
x2 16
x3 16
xcenter (excluded object) 34
xcenter (plate) 29
xcval 16
xlength 29
xmax 29
xmin 29
xo 37
XRay intensity 23
xs 16
xs (camera) 37
Xseed 23

Y

y (collimator) 30
y (cone) 30
y (ellipsoid) 30
y (sphere) 29
y (torus) 33
y (zrprofile) 33
y1 16
y2 16
y3 16
yback (gencoll) 31
ycenter (excluded object) 34
ycenter (plate) 29
yfront (gencoll) 31
ylength 29
ymax 29
ymin 29
yo 37

ys 16
ys (camera) 37
yslice (ellipsoid) 30
yslice (sphere) 29
ysum 34

Z

z (collimator) 30
z (cone) 30
z (ellipsoid) 30
z (sphere) 29
z (torus) 33
z (zrprofile) 33
z1 16
z2 16
z3 16
zbottom 20
zcenter (excluded object) 34
zcenter (plate) 29
zcval 16
zlength 29
zmax 29
zmin 29
zo 37
Zradius (torus) 33
zrprofile 32
zs 16
zs (camera) 37
ztop 20